Complex & Intelligent Systems (2025) 11:264
https://doi.org/10.1007/s40747-025-01851-7

SURVEY AND STATE OF THE ART

Analysis of quantum fully homomorphic encryption schemes (QFHE) and hierarchial memory management for QFHE

Shreya Savadatti[1] · Aswani Kumar Cherukuri[1] · Annapurna Jonnalagadda[2] · Athanasios V. Vasilakos[3,4]

Received: 4 October 2024 / Accepted: 10 March 2025 / Published online: 23 April 2025

Abstract

Homomorphic encryption is a recent and fundamental breakthrough in modern cryptography, which allows the performance of operations on encrypted data without unveiling the data. Leveraging quantum mechanics principles, quantum computers can potentially solve certain computational problems exponentially faster than classical computers. This immense computational power offers new possibilities for various fields, including cryptography. The rapid evolution of both these fields has led to the development of quantum fully homomorphic encryption (QFHE), which makes the capabilities of classical HE extend into the quantum domain. However, many existing QFHE schemes require significant memory due to complex calculations and fault-tolerance needs. This paper contributes in two ways. First, we provide a comprehensive survey of two specific QFHE schemes, discussing their underlying principles, mathematical frameworks, security aspects, and practical applications. We also explore the challenges posed by quantum computing and how QFHE addresses these to achieve both security and computational efficiency. Second, we propose a new hierarchical memory management system for QFHE, which includes a "quantum cache" (a specialized memory storage for quantum data) and a "reinforcement learning agent" (an intelligent system that learns from experience to optimize decisions). This system dynamically manages data movement between the cache and classical memory, improving memory efficiency and potentially boosting computational performance.

Keywords Quantum computing · Homomorphic encryption · Quantum fully homomorphic encryption · Cryptography · Memory management system

Arabic keywords نظام إدارة الذاكرة . التشفير . التشفير الكمومي المتماثل بالكامل . التشفير المتماثل . الحوسبة الكمومية

Introduction

In abstract algebra, homomorphism is defined as a map preserving all the algebraic structures between the domain and range of an algebraic set [1]. The map is simply a function, i.e., an operation, that takes the inputs from the set of domains and outputs an element in the range (e.g., addition, multiplication) [2]. Webster's dictionary defines homomorphism as "a mapping of a mathematical set into or onto another set or itself in such a way that the result obtained by applying operations to elements of the first set is mapped onto the result obtained by applying those corresponding operations to their

Shreya Savadatti, Annapurna Jonnalagadda, and Athanasios V. Vasilakos have contributed equally to this work.

✉ Aswani Kumar Cherukuri
cherukuri@acm.org

✉ Athanasios V. Vasilakos
th.vasilakos@gmail.com

Shreya Savadatti
shreyasavadatti@gmail.com

Annapurna Jonnalagadda
jannapurna@gmail.com

[1] School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore, Tamilnadu 632014, India

[2] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamilnadu 632014, India

[3] Department of Networks and Communications, College of Computer Science and Information Technology, IAU, P.O. Box 1982, 31441 Dammam, Saudi Arabia

[4] Center for AI Research (CAIR), University of Agder (UiA), Grimstad, Norway

respective images in the second set" [3]. In ancient Greek, the term (homos) was used to mean "same," while (morphe) was used for "shape" [4]. Homomorphic encryption (HE) is a significant breakthrough in the field of cryptography, providing a different approach from conventional encryption techniques. At its core, HE enables computation on encrypted data while maintaining the integrity and confidentiality of the underlying information. This is a significant divergence from traditional encryption approaches, which need decoding before any calculation can take place. Traditional encryption requires decryption before computation, which exposes data to security risks, particularly in cloud computing and data-sharing scenarios. Furthermore, conventional encryption methods are inherently limited in their functionality, as they lack the ability to operate directly on encrypted data without first decrypting it [5]. This brings us to HE, a revolutionary development in the realm of cryptology. HE, on the other hand, enables computations directly on encrypted data without the need for decryption, ensuring data privacy and security [5]. The concept of fully homomorphic encryption (FHE) takes the principles of HE further by enabling unlimited computations on encrypted data. FHE schemes allow for both addition and multiplication operations on ciphertexts, which means any arbitrary function can be evaluated on encrypted data without ever needing to decrypt it. This feature opens up new possibilities for secure data processing and storage, making FHE a cornerstone of modern cryptographic research. Despite the immense potential of FHE, it faces significant practical challenges, primarily related to computational efficiency and performance. The operations in FHE schemes tend to be resource-intensive, making them slower and more complex than traditional encryption methods. This has spurred ongoing research aimed at optimizing FHE schemes to make them more viable for real-world applications [6].

Figure 1 illustrates the contrasting workflows of traditional and homomorphic encryption. The graphic depicts how standard encryption systems require data to be decrypted before processing, which can lead to security issues and inefficiencies. In contrast, HE enables calculations to be conducted directly on encrypted data while maintaining confidentiality throughout. This technique enables secure data processing and analysis without exposing sensitive information, providing a considerable advantage for privacy-preserving applications.

The advent of quantum computing has introduced new dimensions and challenges, giving rise to quantum homomorphic encryption (QHE) and specifically, quantum fully homomorphic encryption (QFHE). QFHE extends the principles of classical FHE into the quantum realm, enabling computations on quantum-encrypted data. This new approach not only preserves the security and privacy benefits of classical HE but also leverages quantum computational advantages. QFHE schemes represent a novel and rapidly evolving subset of HE that harnesses the principles of quantum mechanics to enhance data security and computational efficiency. These schemes facilitate operations on quantum states, enabling complex calculations on encrypted quantum data without decryption. The primary components of QFHE include:

- Quantum Key Distribution (QKD): Ensuring secure key exchange using quantum cryptographic principles.
- Quantum Gates: Performing operations on qubits while maintaining their encrypted state.
- Quantum Noise Management: Addressing errors and noise inherent in quantum computations to maintain data integrity

This basic distinction marks a significant advancement in data security and privacy protection. One of the primary benefits of HE is its adaptability. It makes it possible to do a variety of calculations on encrypted data, such as addition, multiplication, and more intricate operations, all without ever disclosing the plaintext that underlies the data. This level of versatility enables applications in a variety of sectors, including secure cloud computing and privacy-preserving data analytics. Furthermore, HE enables secure cooperation and computing across remote datasets, reducing privacy issues and maintaining secrecy throughout the process. This capacity has far-reaching ramifications for areas like healthcare, finance, and telecommunications, where sensitive data must be processed and evaluated while adhering to strict privacy standards.

Quantum fully homomorphic encryption (QFHE) stands as a pivotal breakthrough in cryptography, addressing the dual challenges of security and computational efficiency in the quantum computing era. Traditional FHE has revolutionized secure data processing by enabling encrypted computations, but its significant computational overhead has limited widespread adoption. As quantum computing advances, the threat to classical encryption schemes becomes imminent, necessitating the development of quantum-resilient cryptographic solutions. QFHE builds on the principles of FHE but leverages the unique capabilities of quantum mechanics to enhance both security and performance. By enabling computations on encrypted quantum data without requiring decryption, QFHE ensures privacy even in untrusted quantum environments, such as quantum cloud computing. Its ability to utilize the superior processing power of quantum computers while maintaining end-to-end data security sets it apart from traditional encryption methods. Furthermore, QFHE holds the promise of addressing the computational inefficiencies of classical FHE by reducing latency and resource requirements, making it a practical solution for secure and efficient encrypted computations. Its development marks a critical step toward ensuring robust cryptographic defenses

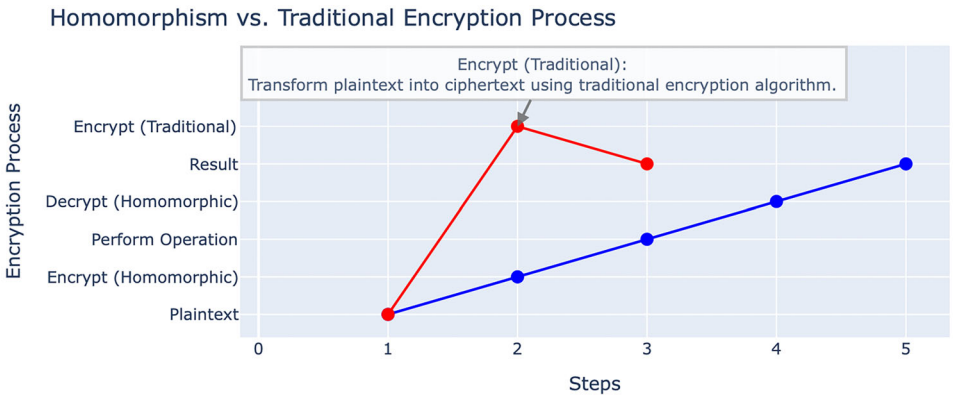

Fig. 1 Illustrates the contrasting workflows of traditional encryption and homomorphic encryption

against both current and future cyber threats, establishing it as an indispensable tool in the evolving landscape of information security.

In addition to the foundational contributions cited in the introduction, recent developments have helped unravel the basic intricacies of QFHE. Particularly, the works of [7–9], presented novel schemes for QFHE and provided insights in several of the mathematical frameworks and efficiency considerations necessary for progress within quantum secure computation.

This paper further aims to contribute to the literature with a profound analysis dedicated only to QFHE. By discussing its functionalities, security properties, and efficiency considerations, together with research challenges, this survey aims to empower researchers and practitioners with tools to explore the emerging paradigm of quantum cryptography confidently. By scrutinizing and critically analyzing, we aim to provide an insight into the potential of QFHE in revolutionizing secure data processing, therefore ushering in a new dawn of quantum-enhanced privacy and confidentiality.

The existing body of literature predominantly focuses on classical FHE, providing minimal coverage of QFHE. Traditional HE surveys have laid a solid foundation for understanding classical schemes, but they often fall short in addressing the unique attributes and complexities of QFHE. Notably, the current surveys:

- Are dominated by traditional FHE methods with little to no information on QFHE.
- Offer general overviews without detailed analyses tailored to QFHE schemes, particularly lacking in-depth examinations of their mathematical frameworks and efficiency considerations.
- Neglect thorough security discussions specific to the quantum realm, leaving potential vulnerabilities inadequately explored.

Recognizing these limitations, this survey carves a unique path by offering a comprehensive and up-to-date analysis dedicated solely to QFHE. To the best of our knowledge, this represents the first thorough examination of QFHE, addressing its functionalities, security properties, efficiency considerations, and open research challenges. The primary objectives of this paper are:

- To provide a detailed analysis of existing QFHE schemes, including their mathematical foundations and performance metrics.
- To examine the specific security considerations and potential vulnerabilities unique to QFHE.
- To identify current research challenges and propose potential directions for future work in the field of QFHE.

By bridging this gap in the literature, we aim to provide valuable insights into the evolving field of QFHE and its implications for the future of secure data computation.

While QFHE has many advantages, current schemes are quite memory-hungry. To make QFHE more practical, we develop a new hierarchical memory management system tailored for QFHE. This system uses an assigned quantum cache and a reinforcement learning (RL) agent to dynamically manage data movement between the cache and classical memory. The RL agent learns access patterns and tunes the placement of data to the minimum extent required to yield the desired performance in computation, minimizing the memory footprint while potentially increasing computation efficiency. This proposed system is further elaborated upon in "Hierarchical memory management with quantum caching and reinforcement learning for QFHE schemes" section.

This paper is structured as follows: "Background" section is dedicated to providing a background. "Fully homomorphic encryption" section will focus on classical FHE, "Quantum computing principles" section proceeds to the principles of quantum computation. "Quantum fully homomorphic schemes" section forms the core part of the paper, fully dedicated to QFHE. Two specific QFHE schemes are taken up for detailed discussion. Then, we discuss the possible applications for the purpose of showing how flexible and

effective QFHE is in practical application. "Hierarchical memory management with quantum caching and reinforcement learning for QFHE schemes" section details our novel hierarchical memory management system designed specifically for QFHE. "Conclusion" section forms the conclusion to our findings.

Background

In the contemporary era of digitization, the significance of data security and privacy has escalated as a result of the substantial volume of confidential data being handled and saved on the internet. Despite the effectiveness of conventional encryption mechanisms in protecting data during storage and transit, they are not suitable for performing operations on encrypted data without prior decryption. Hence, the prominence of FHE comes to the forefront.

Fully homomorphic encryption (FHE) allows for the performance of computations directly on encrypted information, removing the necessity for decryption. This process not only maintains the privacy of the data but also ensures its security throughout the entire computational procedure. The significance of this capability cannot be understated, especially in industries where the confidentiality of data holds paramount importance, such as the healthcare sector, finance domain, and the realm of cloud computing.

The core motivation for the evolution and deployment of FHE is deeply rooted in the sphere of data privacy. Conventional approaches mandate the decryption of data prior to any processing, thereby leaving it vulnerable to potential security breaches. In stark contrast, FHE guarantees that the data remains in its encrypted state throughout the computational operations, effectively shielding sensitive information from any unauthorized access attempts as highlighted by [5]. This aspect is indispensable for upholding the confidentiality of personal data, particularly in the healthcare sector, where safeguarding patient information from prying eyes is of utmost importance as discussed by [10].

FHE is crucial in the financial sector for protecting sensitive financial data. Financial organizations handle massive volumes of personal and transactional data that, if compromised, can cause severe financial and reputational harm. FHE enables banks and financial institutions to do risk analysis, fraud detection, and credit scoring on encrypted data while preventing potential breaches. This ensures that customer data is kept private and secure throughout the processing lifespan. Furthermore, FHE can enable safe multi-party computations, allowing many financial firms to collaborate on encrypted data without disclosing private information [10].

Additionally, the field of secure cloud computing is identified as another pivotal domain where FHE demonstrates substantial utility. With organizations increasingly opting for cloud-based services, there arises a pressing need for ensuring that their data retains its confidentiality even when being processed on servers operated by third-party entities. In this context, FHE emerges as a robust solution by facilitating secure computations on encrypted data within the cloud environment, effectively obviating the necessity of placing blind trust in the security protocols of the service provider, as elucidated by [11].

In addition, FHE supports organizations in abiding by stringent data protection laws like the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA).These provisions demand the safeguarding of personal data throughout its entire life cycle, including the processing phase. FHE enables entities to fulfill these legal obligations by ensuring data confidentiality even during computational processes [12]. FHE also proves useful in protecting intellectual property. Organizations that process proprietary algorithms or data might use FHE to ensure that their intellectual assets are not divulged to cloud service providers or other external actors while computation occurs. This is especially important for maintaining a competitive edge and protecting confidential business methods [13].

Traditional FHE algorithms hold a lot of potential, but they also have certain limitations. One of the primary concerns is the high processing costs and complexity involved. However, these constraints have prompted the development of more efficient methods and implementations. They have also stated that additional improvements in this discipline are required. So, while FHE is undoubtedly a step in the right way, there is still room for growth.

Quantum computing offers the ability to solve some computational problems far more quickly than traditional computers. This poses both opportunities and challenges to our existing encryption systems. Quantum computers may be able to break the widely used cryptographic algorithms on which humans rely, necessitating the development of new encryption methods capable of withstanding quantum attacks. The transition from classical cryptographic algorithms to quantum-resistant techniques has been a critical area of research, particularly in response to the potential risks posed by quantum computing [14]. The article provides a comprehensive overview of the various cryptographic algorithms, highlighting the vulnerabilities of classical encryption methods in the face of quantum capabilities. This transition is crucial in understanding the motivations behind QFHE, which represents a key solution to ensuring data security in a post-quantum world. QFHE, as discussed in [14] allows for computation on encrypted data without decryption, providing a promising defense against the threat of quantum-powered decryption attacks.This has led to the exploration of QFHE. Essentially, traditional encryption techniques face a serious danger from quantum comput-

ing. As quantum computers advance, they could render our present encryption techniques, such as RSA and ECC, completely obsolete. They are simply not prepared to face quantum attacks, such as the infamous Shor's algorithm [15]. But QFHE is intended to be the solution we require. It employs quantum principles to provide protection against quantum attacks, allowing us to stay one step ahead [16].

In summary, QFHE is a means by which we may secure our encryption techniques against the threat posed by quantum computers and ensure their continued viability. Furthermore, quantum computers have the ability to perform FHE operations more efficiently than conventional computers, hence lowering the computational overhead and latency associated with present FHE methods. This makes QFHE an appealing choice for performing sophisticated encrypted computations in a quantum computing environment [17].QFHE, like classical FHE, can enable secure data processing in a quantum cloud computing environment. This means that sensitive data can be processed by quantum servers without being disclosed, guaranteeing that the data stays private even when using quantum computers' increased processing capacity. The development of QFHE schemes is critical for developing secure cryptographic approaches that will resist the future landscape of quantum computing. It is crucial to guarantee robust security in a quantum future to maintain long-term data security and privacy, as well as to keep ahead of possible risks offered by quantum advancements. [18].

However, the cost of quantum computing remains a significant barrier to its widespread adoption and practical implementation. Building and maintaining quantum systems involves substantial financial investment due to their technical complexity and the advanced infrastructure required. For instance, most quantum computers operate at near absolute zero temperatures, necessitating the use of costly cryogenic systems. Moreover, the high-precision hardware needed, such as error-correcting mechanisms and quantum gates, adds to the expense. Beyond the physical systems, the scarcity of specialized expertise to design, program, and optimize quantum technologies further drives up costs. Ongoing research and development in quantum computing, aimed at enhancing scalability and reliability, also demand significant funding. These factors collectively make quantum computing an investment-intensive domain, with costs spanning infrastructure, talent, and software development. While QFHE presents a compelling vision for secure quantum cloud computing, addressing the financial barriers associated with quantum systems will be critical to unlocking its full potential and broader accessibility.

To summarize, although FHE meets the immediate need for secure computation on encrypted data, the growth of quantum computing necessitates the exploration and development of QFHE to provide strong security in the quantum future. These developments are crucial for ensuring data privacy and security in an ever-changing technological context.

Fully homomorphic encryption

FHE schemes

Fully homomorphic encryption (FHE) has had a significant impact on cryptography since it permits computations on ciphertext without requiring decryption. A key contribution in this area is the lattice-based FHE scheme put forward by Craig Gentry in 2009. Gentry's scheme uses ideal lattices and the bootstrapping technique to manage the noise growth problem caused by homomorphic operations, allowing computations on cipher texts to be done indefinitely [5]. Gentry's scheme can be summarized as follows:

Encryption: The encryption function $\mathrm{Enc}(m)$ maps a plaintext m to a ciphertext c.

Homomorphic Operations: Given two ciphertexts $c_1 = \mathrm{Enc}(m_1)$ and $c_2 = \mathrm{Enc}(m_2)$, it is possible to compute $\mathrm{Enc}(m_1 + m_2)$ and $\mathrm{Enc}(m_1 \cdot m_2)$ directly from c_1 and c_2 without decrypting them.

Decryption: The decryption function $\mathrm{Dec}(c)$ maps a ciphertext c back to the plaintext m.

Mathematically, if $c_1 = \mathrm{Enc}(m_1)$ and $c_2 = \mathrm{Enc}(m_2)$, then:

$$\mathrm{Enc}(m_1 + m_2) = c_1 \oplus c_2$$
$$\mathrm{Enc}(m_1 \cdot m_2) = c_1 \odot c_2$$

LWE-based HE scheme offers several advantages over Gentry's original scheme from 2009, including conceptual simplicity and asymptotic speed improvements. It has implications for various applications in secure computation, privacy-preserving data analysis, and cloud computing.

Key Generation: Generate a public-private key pair (pk, sk) using the LWE problem.

Encryption: Encrypt a plaintext m using the public key pk to produce a ciphertext c by adding noise sampled from a certain distribution. Mathematically, it can be represented as: $c = \mathrm{Enc}(m)$.

Homomorphic Operations: Perform homomorphic addition and multiplication operations on ciphertexts. Given two ciphertexts $c_1 = \mathrm{Enc}(m_1)$ and $c_2 = \mathrm{Enc}(m_2)$, the homomorphic addition and multiplication can be represented as:

$$c_3 = c_1 + c_2$$
$$c_4 = c_1 \times c_2$$

Decryption: Decrypt the resulting ciphertext c_3 or c_4 using the private key sk to obtain the plaintext result.

Subsequent schemes have been built to improve the efficiency and practicality. For instance, the Brakerski-Gentry-Vaikuntanathan (BGV) scheme further simplifies the bootstrapping process and minimizes the computational overhead by leveraging ring learning with errors (RLWE) as the hardness assumption in the underlying lattice problem [10]. We provide here a generic and simple description of the BGV scheme in the following steps:

Key Generation: Generate a public-private key pair (pk, sk).

Encryption: Encrypt a plaintext m using the public key pk to produce a ciphertext c.

Evaluation: Perform homomorphic operations on ciphertexts. For example, given $c_1 = \text{Enc}(m_1)$ and $c_2 = \text{Enc}(m_2)$, compute $c_3 = \text{Eval}(f, c_1, c_2)$ where f is a function.

Decryption: Decrypt the resulting ciphertext c_3 using the private key sk to obtain the plaintext result.

Besides, the CKKS encryption scheme gives significant support to approximate HE. This is vital to performing machine learning and scientific computations over the real numbers even while operating in the encrypted domain [19]. In essence, the CKKS scheme can be summed up as follows:

Encryption: Encrypts a plaintext vector m into a ciphertext c.

Homomorphic Operations: Supports addition and multiplication of encrypted vectors, allowing for operations on encrypted data without decryption.

Decryption: Decrypts the resulting ciphertext back into a plaintext vector.

FHE libraries

Many libraries have been developed to facilitate the implementation and application of the FHE schemes. Amongst them is HElib, an open-source library by IBM Research implementing the BGV scheme, designed for performance and flexibility, with a solid framework for several cryptographic applications [20]. Some of the features in this library include the following:

1. Optimized arithmetic operation, both for large integers and for polynomials.
2. Parameter selection, based on secure and effective encryption.

Another major library is SEAL-Microsoft's Simple Encrypted Arithmetic Library, which implements the BFV (Brakerski/Fan-Vercauteren) as well as the CKKS schemes. The reason it is considered easy to use is largely due to its comprehensive documentation and is used by developers as well as researchers [21]. Some of the functions within it include the following:

1. Homomorphic arithmetic functions with addition and multiplication for encrypted data.
2. Serialization and deserialization of encrypted data, to support storage and transmission.

The paper [22] describes numerous C++ libraries for HE, such as OpenFHE, TFHE, and HEAAN. OpenFHE supports a variety of schemes, including BGV, BFV, CKKS, DM, and CGGI, and is used for secure machine learning, privacy-preserving data analysis, and secure cloud computing. TFHE is known for its efficiency when compared to other FHE methods, and it is used in secure machine learning and privacy-preserving data analytics. HEAAN focuses on computations with approximate numbers and is utilized in similar situations.

The FHE landscape is influenced by the prominent PALISADE library, which is a collaboration between academic and industry partners. It plays a significant role by offering various FHE schemes, such as BGV, CKKS, and others, with an emphasis on modularity and efficiency. PALISADE enables users to explore different cryptographic settings and parameters, thereby facilitating research and development in HE [23]. PALISADE boasts several key features, including:

1. It supports multiple encryption schemes and cryptographic primitives.
2. It utilizes techniques to enhance performance on different hardware platforms.

TFHE is a library developed by Zama to run FHE computations on hardware accelerators like GPUs and TPUs. It really shines when it comes to hardware acceleration in applications that demand high performance and large datasets [24].

Lattigo is a Go library focusing on implementing HE schemes over lattices. It is designed to present a user-friendly interface and is geared toward efficiency and scalability in practice. Lattigo is the go-to choice for lightweight FHE libraries and ensuring top performance [25].

In their paper [26] compare the performance of various HE libraries, specifically focusing on BFV implementations in SEAL and PALISADE. They assessed five key operations (KeyGen, Enc, Dec, Add, and Mult) using consistent parameters across both libraries. The results reveal that while SEAL generally exhibits faster execution times, particularly for multiplication, the performance of both libraries is comparable overall. SEAL's advantage in multiplication is attributed to its separate handling of relinearization compared to PAL-

ISADE, which incorporates this step within its multiplication function.

Future directions and challenges

While FHE is promising, there are several challenges that it poses, and improvements need to be made for it to reach its potential. High computational overhead is the main obstacle to full-blown FHE. Although a lot of progress has been made in this line of optimization in homomorphic operations, more has to be done to make FHE practical in applications involving large datasets and complicated computations.

Another is noise growth in ciphertexts. Although such an effect has been reduced to some limit via bootstrapping techniques, they impose extra overhead. Current research is aiming at developing more efficient noise management techniques to improve the performance of FHE schemes further [27].

Future research on FHE is likely to focus on enhancing efficiency, scalability, and usability. It also requires schemes that are resistant to quantum computing, as the threat of quantum computing advances. As these challenges are addressed, FHE is expected to play an increasingly vital role in secure data processing and privacy-preserving technologies.

Quantum computing principles

The utilization of quantum mechanical ideas to perform computation on data is the definition of quantum computing. A potentially very powerful new computer paradigm, quantum computing is promising revolutionary breakthroughs in useful areas such as material science, cryptography, optimization, and more. Basic ideas behind quantum computing: Qubits, superposition, entanglement, and quantum gates.

Quantum bits (Qubits)

Unlike classical bits that exist in a state of 0 or 1, qubits can exist in a superposition of both states simultaneously. Mathematically, a qubit's state is represented as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where α and β are complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$. This superposition enables quantum computers to process a vast amount of information simultaneously, which is pivotal for their computational power [28].

Superposition

Superposition is a fundamental principle that allows qubits to be in multiple states at once. For instance, if a qubit is in a superposition state $|\psi\rangle$, it can be expressed as a linear combination of its basis states $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

This principle is what gives quantum computers their exponential parallelism, making them capable of solving certain problems much faster than classical computers [29].

Entanglement

Entanglement is a signature quantum feature where qubits become interdependent, in such a way that the state of one qubit directly affects the state of another, regardless of separation distance. For instance, for an entangled pair of qubits in the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle),$$

instantaneous measurement of the first qubit determines the state of the second qubit. It is this property that is central to many quantum algorithms and protocols, including the teleportation of quantum states and superdense coding [30].

Quantum gates

Quantum gates are used in a quantum circuit in the same way that logic gates are used in classical circuits. These transformations are accomplished unitarily and preserve the norm of the state vector of qubits, being described by unitary operators. Some commonly used quantum gates are the Pauli-X, Hadamard, and CNOT gates.

Pauli-X gate

The Pauli-X gate, also known as the quantum NOT gate, flips the state of a qubit. It is represented by the following unitary matrix:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

The action of the Pauli-X gate on the basis states is:

$$X|0\rangle = |1\rangle$$
$$X|1\rangle = |0\rangle$$

Pauli-Y gate

The Pauli-Y gate introduces a phase flip along with the bit flip. It is represented by the matrix:

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

The action on the basis states is:

$$Y|0\rangle = i|1\rangle$$
$$Y|1\rangle = -i|0\rangle$$

Pauli-Z gate

The Pauli-Z gate, also known as the phase-flip gate, inverts the phase of the qubit. It is represented by:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The action on the basis states is:

$$Z|0\rangle = |0\rangle$$
$$Z|1\rangle = -|1\rangle$$

Hadamard gate

The Hadamard gate (H) creates a superposition of states. It is represented by:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

The transformation of the basis states is:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

CNOT gate

The Controlled-NOT (CNOT) gate is a two-qubit gate that flips the state of the second qubit (target qubit) if the first qubit (control qubit) is in the state $|1\rangle$. The matrix representation of the CNOT gate is:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

The action of the CNOT gate can be described as:

$$\text{CNOT}|00\rangle = |00\rangle$$
$$\text{CNOT}|01\rangle = |01\rangle$$
$$\text{CNOT}|10\rangle = |11\rangle$$
$$\text{CNOT}|11\rangle = |10\rangle$$

Phase gate

The phase gate (S) shifts the phase of the state $|1\rangle$ by $\pi/2$. It is represented as:

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

The action on the basis states is:

$$S|0\rangle = |0\rangle$$
$$S|1\rangle = i|1\rangle$$

T gate

The T gate is a phase shift gate that shifts the phase of the state $|1\rangle$ by $\pi/4$. It is represented as:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

The action on the basis states is:

$$T|0\rangle = |0\rangle$$
$$T|1\rangle = e^{i\pi/4}|1\rangle$$

Swap gate

The Swap gate exchanges the states of two qubits. It is represented by the matrix:

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The action on the basis states is:

$$\text{SWAP}|ab\rangle = |ba\rangle$$

These gates, when combined, form the basis for constructing complex quantum algorithms and circuits capable of solving problems in ways that classical computers cannot match [28, 31].

Quantum algorithms

Shor's algorithm

One of the most famous quantum algorithms is Shor's algorithm for factoring large integers, with strong implications for cryptography. While classical algorithms to factor are exponentially slow, Shor's algorithm factors an integer N in

polynomial time. The algorithm uses quantum Fourier transform and is as follows:

1. Initialize a superposition of all possible states.
2. Apply the quantum Fourier transform.
3. Measure to find the period of a function related to the factors of N.

This results in the efficient discovery of the factors of N [32]. The quantum Fourier transform (QFT) is defined through its action on the computational basis states: The quantum Fourier transform (QFT) on a state $|k\rangle$ is defined as:

$$\text{QFT}|k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi ikj/N} |j\rangle$$

where N is the dimension of the Hilbert space.

Grover's algorithm

Grover's algorithm offers a quadratic speedup for unstructured search problems. For a database of N entries, Grover's algorithm finds a target entry in $O(\sqrt{N})$ time, compared to $O(N)$ for classical algorithms. The steps of the algorithm is:

1. Prepare a superposition over all possible states.
2. Perform the Grover iteration, which is composed of the Oracle and the diffusion operator.
3. Iterate over the operation $O(\sqrt{N})$ times, in order to amplify the probability of the right state.

This will measure the desired state with high probability [33] Mathematically, Grover's iteration is represented as:

$$G = (2|\psi\rangle\langle\psi| - I)\, O$$

where O is the oracle that marks the solution state and $|\psi\rangle$ is the initial equal superposition state.

Quantum error correction

Decoherence and other quantum noises represent a great source of errors in quantum computers. It is of utmost importance to come up with ways of correcting quantum errors for quantum computers to be efficient. One of the most well-known codes for this is the Shor code. It encodes one qubit into nine qubits and can correct arbitrary single-qubit errors. The Shor code is based on both the bit-flip and phase-flip code. The logical qubits $|\psi\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$ are encoded into nine physical qubits, represented as:

$$\alpha|0_L\rangle + \beta|1_L\rangle$$

where $|0_L\rangle$ and $|1_L\rangle$ are logical qubits encoded into nine physical qubits [34].

Quantum fully homomorphic schemes

Quantum fully homomprhic encryption

Quantum fully homomorphic encryption (QFHE) is among the landmark achievements in cryptographic theory that ensures incomparable security in computations on encrypted data. Although the practicality of classical FHE schemes has been founded, the schemes are really held back by computational overhead and vulnerability to noise. Leveraging quantum mechanical properties, QFHE promises to go past these difficulties and provide a way of attaining efficient and secure computation on encrypted quantum states.

The key primitive of QFHE lies in the ability to evaluate quantum circuits on encrypted quantum data. Unlike classical FHE which operates on classical data, QFHE encrypts quantum states, enabling computation to be performed directly on the encrypted states. This not only ensures confidentiality but also dispenses with the need for decryption, leading to reduced exposure risks.

Pioneering work by [35] developed the first scheme for QFHE, specializing in circuits with low T-gate complexity. It is based on the use of quantum entanglement and superposition to perform computations homomorphically, preserving data integrity, thereby enabling seamless evaluation.

QFHE is mathematically grounded in quantum information theory, and uses formalisms such as Hilbert spaces, quantum operations, and entanglement measures. Denote by H the Hilbert space, representing quantum states, and by $\mathcal{L}(H)$ the set of linear operators on H.

As shown in Fig. 2, the core algorithms involved in QFHE include Key Generation (Gen), Encryption (Enc), and Homomorphic Evaluation (Eval). This diagram provides a visual representation of the process and interactions between these components.

A QFHE scheme consists of the following algorithms:

- Key Generation (**Gen**): This algorithm generates a public key, pk, and secret key, sk. $(pk, sk) \leftarrow \text{Gen}$
- Encryption (**Enc**): It takes a quantum message, m, and a public key, pk, and the encryption algorithm **Enc** generates the encrypted quantum state, ρ. $\rho = \text{Enc}(m, pk)$
- Homomorphic Evaluation (**Eval**): This algorithm allows one to compute quantum circuits on encrypted quantum states. It will take the encrypted state, ρ, and the quantum circuit, C, and **Eval** returns the encrypted state, ρ', representing the result of applying the quantum circuit, C, to the state, ρ. $\rho' = \text{Eval}(\rho, C)$

The foundation of QFHE is predicated on the intricate nature of quantum computational problems, as exemplified by the Quantum Hidden Subgroup Problem. This cryptographic scheme relies on the advanced capability to maintain coherence during computation, thereby reducing the risks associated with potential eavesdropping activities. Moreover, the successful implementation of QFHE requires the incorporation of robust error correction mechanisms to effectively address the intrinsic noise and decoherence present in quantum systems.

The essence of QFHE is deeply rooted in the amalgamation of quantum mechanics and cryptography, promising cutting-edge advancements in privacy and security. The achievement of Quantum FHE, nevertheless, hinges on overcoming numerous technical hurdles and enriching the realm of quantum computation. Consequently, this endeavor demands extensive research efforts to address these challenges and propel the field forward towards practical implementation.

Quantum fully homomprhic encryption scheme

In the context of QFHE schemes, we have chosen two very different constructions, namely quantum fault-tolerant construction based encryption scheme [36] and quantum one-time pad (QOTP)-based scheme [37]. Quantum fault-tolerant based scheme relies on classical error-correction techniques, particularly CSS codes, and leads to a very compact and scalable QFHE construction. On the other hand, QOTP-based scheme is based on the combination of QOTP and the classical FHE approach to secure computation over encrypted data.

The choice of these two schemes will allow one to cover a broad set of aspects related to QFHE research. They differ first in the kind of security they target: while quantum fault-tolerant based scheme achieves perfect security by the size of the CSS code set, the security of QOTP-based scheme is based on the hardness of the Learning With Errors (LWE) problem, aiming at computational security. This difference will allow a fine discussion of security notions in QFHE, enriched with mathematical rigour and cryptographic depth.

Furthermore, the strengths and weaknesses of the two schemes offer a reader valuable insights into the trade-offs between efficiency and security in the design of a practical QFHE scheme. The compactness and scalability of quantum fault-tolerant based scheme can be used in handling large circuits, but it does require fault-tolerant implementation of the basic quantum gates for reliable computation. On the contrary, QOTP-based scheme performs efficient computation based on HE but might suffer in its practical applicability under some settings due to circuit depth and the number of qubits.

Both schemes have been presented at highly reputable conferences, and therefore may be interesting as they present highly important results in improving QFHE research.

Going forward, we will follow this with a general review of both schemes, which highlights their underlying ideas and cryptographic concepts. In the following section, we perform a comparative security analysis between the two schemes based on perfect security and computational security. We also compare the evaluation efficiency of the two schemes with respect to key generation, encryption, decryption, and gate evaluation. The practical considerations taken into account in this comparison are fault-tolerant implementations and circuit complexity limitations. Finally, we summarize the main differences between the two schemes and the applications for which each would be best suited in order to capture the essence of our comparative analysis.

Overview

Quantum fault-tolerant based scheme presents a novel approach in the field of QFHE, using quantum fault-tolerant construction to increase security and efficiency in cryptographic protocols. This approach marks a significant advancement in the realm of quantum computing, paving the way for enhanced security measures and enabling complex computations to be performed on encrypted data without compromising the confidentiality of the information. It carefully presents both symmetric and asymmetric schemes, each carefully designed to solve specific cryptographic issues while maintaining data privacy and security. It offers advantages like security based on the difficulty of breaking the underlying CSS code and fault-tolerant evaluations, making it a valuable contribution to the field. However, the use of an ancillary state and the symmetric nature of the scheme are aspects to consider for further improvement.

The cornerstone of the symmetric scheme lies in utilizing quantum CSS codes for the secret key. CSS codes are commonly known for their proficiency in fault-tolerant quantum computation, forming the basis for the framework of the encryption. CSS codes function similarly to error-correction protocols, enabling the encoding of quantum states in a manner that facilitates smooth operations on physical qubits without the necessity for ad hoc error-correction protocols. Through fault-tolerant techniques, the H, CNOT, and S gates can be executed precisely on physical qubits, thereby enhancing computational reliability without the burdensome use of error-correction procedures. The encoding of quantum states using CSS codes makes the scheme possible. It allows logical qubit operations seamlessly through the transversal operation on physical qubits, hence ensuring strong fault tolerance. In particular, the integration of an ancillary state intricately linked to the CSS code plays a crucial role in executing non-commutative quantum gates, specifically the T

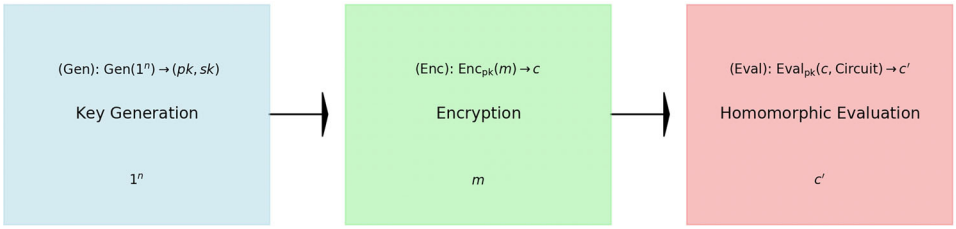

Fig. 2 This diagram depicts the core algorithms involved in QFHE: Key Generation (Gen), Encryption (Enc), and Homomorphic Evaluation (Eval)

gate. This symmetric scheme presents evidence of a strong encryption paradigm; it does not require the usual periodic error-correction procedures and thus strongly resists the conventional cryptographic challenges.

Quantum CSS Codes (Q_n): The symmetric scheme is based on quantum CSS codes with the secret key (sk). These codes are given by Q_n, and they form the building blocks for fault-tolerant quantum computation. Mathematically, $|Q_n| = 2^n - 1$, where n is the security parameter.

CSS Encoding: The quantum plaintext ρ chosen to be sent is encrypted using the chosen CSS code, denoted as (sk). As a result, the output is an encrypted state $\sigma = \text{Enc}_{sk}(\rho)$.

Ancillary State ($|\Theta\rangle$): In order to facilitate non-commutative quantum gates, especially the T gate, an ancillary state that depends on the CSS code (sk) to be used is defined. The state is given mathematically as:

$$|\Theta\rangle = \text{Enc}_{sk}\left(|0\rangle + e^{i\pi/4}|1\rangle\right)$$

Fault-Tolerant Computation: Bob, in the role of the receiver, performs fault-tolerant quantum computation (QC) on the encrypted state (σ). The output is the encoded state $\sigma' = QC(\sigma)$, which accomplishes secure computation without requiring decryption.

CSS Decoding: The decryption of the result, in this case, involves decoding the encrypted result (σ') using the secret key (sk). Thus, the resulting plaintext is $\rho' = \text{Dec}_{sk}(\sigma')$.

On the other hand, the asymmetric scheme addresses this non-commutativity between the CSS encoder and some specific quantum operators, such as the T gate. The introduction of random Pauli errors during encryption effectively solves this challenge at the same time, reinforcing computational security. Additionally, with the seamless integration of an interactive process between Alice and Bob, the scheme manages error propagation expertly, ensuring the integrity and confidentiality of the encrypted data. This scheme emphasizes not only its exploration potential, specifically in quantum bootstrapping, but also the drive toward the harmonization of encryption procedures while simultaneously minimizing the overhead resulting from interaction.

At the core of the success of both schemes is the innovative application of quantum fault-tolerant construction. This base framework not only enables secure computation on encrypted data but also boosts scalability and efficiency in quantum encryption protocols. These schemes also provide invaluable insight into the delicate balance of quantum error correction, gate operation, and encryption upon which the ability to conduct quantum cryptography and secure multiparty computation in a scalable fashion stands.

QOTP-based scheme presents a QFHE design that practically matches the security provided by classical systems. Building on the work of [38], the scheme manages to achieve exponentially diminished per-gate error rates in a unique way that interlocks quantum encryption with classical lattice-based cryptography.

At the core of the scheme is key encapsulation, with the primal [39–41] lattice-based scheme as the Key Encapsulation Mechanism (KEM). In contrast to classical lattice-based schemes, This scheme does not require a lattice trapdoor as part of its secret key. However, its secret key consists of an encryption of the secret key, and the randomness that was used to generate that encryption.

An important feature of this scheme is the ability to homomorphically evaluate quantum circuits. Namely, the scheme can support the homomorphic evaluation of quantum cryptographic functions. It bases this ability on classical notions of privacy, adapting techniques for privacy-preserving classical computation to the quantum setting. The scheme depends on circuit privacy, meaning that a homomorphically evaluated function on the input state hides the function that was evaluated. This property, which was previously considered a security measure, now proves to be essential for the verification of the correctness of homomorphic quantum evaluations.

The design requires the use of advanced quantum operations, such as QOTP encryption and homomorphic evaluation of classical circuits over quantum ciphertexts. The crucial idea in this work is to introduce randomness in the encryption that preserves privacy, but still allows homomorphic processing over quantum ciphertexts. The scheme further allows one to homomorphically evaluate CNOT operations, a feature that is in the core of quantum computations. The scheme can support such computations since it makes use of randomness in the encryption while keeping lattice-based techniques. It guarantees privacy and allows correctness even in the quantum setting.

This scheme combines classical lattice-based cryptography with quantum encryption methods, leading to exponentially reduced per-gate errors. The following are the main

mathematical concepts and equations used in achieving this result.

Key Encapsulation Mechanism (KEM): The KEM component uses the primal lattice-based scheme of [GSW13, BV14, AP14]. Let A be a public key matrix and s be a secret key vector. The encryption of a bit x is a matrix C of the following form:

$$C = AR + xG \pmod{q}$$

where R is a matrix of small norm, G is some special "gadget" matrix, and q is a modulus. This scheme is secure based on the Learning With Errors (LWE) assumption, with χ as the error distribution.

Circuit privacy: This scheme possesses the feature of circuit privacy, which implies that homomorphically evaluated quantum functions are hidden from an outside observer. More precisely, after homomorphic evaluation, the resulting ciphertext $\mathrm{Enc}(f(x))$ should reveal no more information about f (other than $f(x)$).

Quantum one-time pad (QOTP): The QOTP encryption of a qubit x is that of a random Pauli operation - a random bit flip and a random phase flip. This operation can be used to evaluate Clifford gates in quantum circuits.

CNOT operations: One of the main new features of the scheme is the ability to homomorphically evaluate CNOT operations acting on a quantum state given an encryption of a classical control bit. More precisely, given a 2-qubit superposition $P_{a,b}\alpha_{a,b}|a, b\rangle$ and an encrypted control bit x, the scheme evaluates an encapsulated encryption of $P_{a,b}\alpha_{a,b}|a, b \oplus ax\rangle$.

Mathematical formulation: The functionality of the scheme is based on rigorous mathematical formulations ensuring both security and correctness in quantum computations. The lattice-based cryptographic methods together with quantum encryption principles allow this achievement of strength in HE in the quantum domain.

Security analysis

Quantum fault-tolerant based scheme offers two flavors of QFHE: symmetric and asymmetric. Table 1 compares the security of asymmetric and symmetric schemes. The symmetric system offers perfect security through CSS code, whereas the asymmetric scheme relies on computational security based on an unspecified hard problem. The table also highlights how each scheme handles faults and ancillary states, emphasizing the symmetric scheme's vulnerability to ancillary state exposure and the asymmetric scheme's interactive error correction method.

QOTP based scheme offers a practical approach to achieving computational security for transmitting quantum information. Although the QOTP provides flawless secrecy, it relies on a cryptographic hash function (CHE) that is dependent on the Learning With Errors (LWE) assumption.

It is very important to select the right parameters of the LWE problem to use within the CHE scheme: these parameters determine the difficulty of breaking the encryption and the noise level of the ciphertexts. Fresh, unpredictable randomness in the generation of the one-time pad (quantum state) is crucial for security reasons. Furthermore, the post-quantum digital signature schemes can be incorporated for the authentication of the sender and integrity of the message. Furthermore, one should remain alert to the quantum channel at all times in order to detect a probable eavesdropper and prevent a security compromise. Table 2 summarizes the security analysis of a QOTP-based system. It discusses the scheme's reliance on computational security via Learning With Errors (LWE) and the use of shared randomness for the QOTP, which ensures perfect secrecy. The security is maintained by combining QOTP, which theoretically provides perfect secrecy, with Classical Homomorphic Encryption (CHE), which is based on the hardness of LWE. Depending on the CHE scheme employed, potential leakage may reveal information about the QOTP key.

Efficiency

Quantum fault-tolerant based scheme presents a QFHE scheme based on quantum fault-tolerant construction, encompassing both symmetric and asymmetric variants. This analysis delves into the efficiency of key generation, encryption, decryption, and gate evaluation, highlighting critical factors that influence performance. QFHE allows computations directly on encrypted quantum data. At the heart of this computation capability, there exists a critical operation: gate evaluation. It includes applying quantum logic gates; for instance, Hadamard, CNOT; to the encrypted quantum state representing the message. These gates operate on the qubits in the encrypted state according to the desired computation, making transformations on the data that is under it, without ever decrypting it.

Efficiency in key generation, encryption, decryption, and gate evaluation are features sought in a QFHE scheme, especially in the light of fault tolerance. Efficient key generation is important to make the scheme practical for different applications. If key generation is computationally expensive, it can compromise the practical use of the QFHE system. The efficiency in encryption describes the time and the resources necessary for the cryptographic protection of messages. It would be pleasing if this operation could be fast enough for everyday communication and not lead to significant delays. Typically, decryption is much less frequent than encryption in QFHE. Still, efficient decryption should be there to retrieve the original message in due time when neces-

Table 1 Security analysis comparison with use case

Property	Symmetric scheme	Asymmetric scheme
Security Type	Perfect security (information-theoretic)	Computational security (based on a hard problem)
Secret Key	A Quantum Error-Correcting Code (CSS code) - sk	Triple (S, G, P) used to generate a CSS code
Security Analysis	Relies on the hardness of breaking the CSS code. An attacker cannot decrypt the ciphertext without knowing the specific CSS code used	Achieves security through the NP-completeness of a yet unspecified problem. The paper mentions this but leaves the formal details for a full version
Security vs. Ancillary States	The T gate operation requires an ancillary state, $\|\Theta\rangle$, which depends on the secret CSS code. An attacker cannot decrypt the ciphertext without knowing the specific CSS code used. This introduces a vulnerability: If the attacker can obtain many ancillary states, they might gain information about sk. Mitigating this risk: Alice should choose a large enough security parameter (n) for the CSS code to make successful attacks exponentially unlikely	Encryption adds random errors to the encoded state. To prevent error propagation during computations, the scheme adopts an interactive approach: Bob performs a part of the computation. If errors reach a threshold, Bob sends the intermediate result to Alice. Alice decrypts, re-encrypts, and sends it back for Bob to continue. This interaction ensures errors remain manageable
Use Case	A good use case for symmetric schemes is in situations where two parties already trust each other, like in secure quantum communication. These schemes are ideal because they provide perfect security and ensure that data sent between them remains safe from hackers	Asymmetric schemes are better suited for situations where two parties don't know each other in advance, such as in secure online communication. These schemes rely on computational security, which makes them a good choice for applications like securing data on the cloud, where key management needs to be more flexible and scalable

Table 2 Security analysis of QOTP based scheme

Property	Description
Security type	Computational security (LWE-based)
Secret Key	None (shared randomness used for QOTP)
Security Analysis	Scheme 21 achieves security by combining two components
	Quantum One-Time Pad (QOTP): This theoretically guarantees perfect secrecy. An adversary cannot decrypt the message without knowing the secret one-time pad (quantum state)
	Classical Homomorphic Encryption (CHE): This encrypts the key used for the QOTP. The security of CHE relies on the hardness of the Learning With Errors (LWE) problem. This means an adversary with limited computational resources cannot break the scheme within a reasonable time frame
Security vs. Leakage	Leakage from CHE: The security of the scheme depends on the chosen CHE scheme. Leakage from the CHE scheme, if any, could potentially reveal information about the key used for the QOTP

sary. If discussing efficiency regarding fault tolerance QFHE, Gate evaluation is possibly the most important one. In the fault-tolerant evaluation of gates, error correction operations usually need to be computed. For this reason, these operations often result in massive overheads. A typical example analyzes the efficiency of gate evaluation, which would ensure that the computations on the encrypted data are practical within reasonable time and resource bounds.

Table 3 compares the efficiency of symmetric and asymmetric methods, with emphasis on key generation, encryption, decryption, and gate assessment. Both systems have polynomial complexity for key generation and encryption, with symmetric schemes boosting qubit count significantly and asymmetric schemes adding small cost due to random mistakes. Gate evaluation is efficient in both schemes; however, symmetric schemes require additional contact for T gates, whereas asymmetric schemes handle error buildup by periodic interactions. For evaluating the efficiency of a QFHE scheme, especially a scheme based on quantum fault-tolerant construction, more factors must be added to get a more thorough analysis. Here are a few new factors that one can consider:

- Communication Overhead:
  - Symmetric Scheme: The possible communication overhead will stem in the need for ancillary states for the T-gates. How often and how much data is passed between Alice and Bob determines the overall efficiency.
  - Asymmetric Scheme: Periodic interaction between Alice and Bob for error correction leads to more sig-

Table 3 Efficiency comparison between symmetric and asymmetric schemes

Property	Symmetric scheme	Asymmetric scheme
Key Generation Efficiency	1. Process: Alice selects a quantum CSS code from a set Q_n of 2^{n-1} possible codes. The secret key can be a generator matrix G transformed using random nonsingular binary matrix S and permutation matrix P	1. Process: Alice determines a private key tuple (S, G, P) and computes $\hat{G} = SGP$
	2. Efficiency: The key generation involves choosing and transforming matrices, which is efficient for moderate n. The complexity is polynomial in the size of the matrix, specifically $O(n^2)$	2. Efficiency: Similar to the symmetric scheme, generating and transforming matrices involve polynomial complexity. The additional step of calculating $\hat{G}$ also remains within $O(n^2)$
Encryption Efficiency	1. Process: Alice performs CSS encoding on quantum plaintext ρ to obtain $\sigma = \text{Enc}_{sk}(\rho)$, encoding each qubit into n qubits	1. Process: Similar to the symmetric scheme, but with added random Pauli errors
	2. Efficiency: Encoding involves applying stabilizer codes, which is efficient and has polynomial complexity. However, the overhead is significant, increasing qubit count by a factor of n, leading to a total qubit count of $n\times$ the number of logical qubits	2. Efficiency: Encoding with additional errors slightly increases complexity but remains polynomial. The overhead of adding errors is $O(n)$
Decryption Efficiency	1. Process: Alice decodes the quantum ciphertext σ' to obtain the plaintext $\rho' = \text{Dec}_{sk}(\sigma')$	1. Process: Alice decodes the state returned by Bob periodically during evaluation
	2. Efficiency: Decoding is a reverse of encoding, involving stabilizer decoding, with polynomial complexity. The overhead is similarly $O(n)$ per qubit	2. Efficiency: Decryption with periodic interaction involves additional computational steps but each step remains polynomial. The overhead is the communication and decoding steps, adding to the interaction cost
Gate Evaluation Efficiency	1. Process: Bob performs fault-tolerant quantum computation QC on encoded data. Fault-tolerant gates (H, CNOT) are implemented transversally	1. Process: Similar fault-tolerant operations, but with random errors and periodic interaction for error correction
	2. Efficiency: Transversal gates are efficient, with a complexity of $O(n)$ per logical gate. However, T gates require ancillary states from Alice, adding interaction overhead	2. Efficiency: Evaluation is efficient for H and CNOT gates ($O(n)$). T gates are managed through interaction, increasing complexity but maintaining polynomial bounds
	3. Circuit Depth and Qubit Count Impact: The depth of the circuit increases linearly with the number of T gates due to the need for ancillary states. Qubit count is multiplied by n for fault tolerance	3. Circuit Depth and Qubit Count Impact: Interaction periods control error growth, limiting circuit depth impact. Qubit count increases with added errors but remains polynomially bounded
Factors Impacting Performance	1. Circuit Depth: Directly influenced by the number of T gates, requiring ancillary states	1. Circuit Depth: Managed through periodic interaction, controlling error accumulation
	2. Qubit Count: Scales linearly with the size of the CSS code n	2. Qubit Count: Slightly higher due to added Pauli errors but similarly scales linearly with n
Use Case	1. In practical applications, symmetric schemes are particularly suited for scenarios where two trusted parties, such as in secure communication between two quantum computers or quantum key distribution systems, need to exchange sensitive information	1. On the other hand, asymmetric schemes excel in more dynamic and scalable environments where communication needs to happen between multiple, untrusted parties
		2. For instance, in secure cloud-based quantum computing, multiple users access a shared quantum server to process their data, without any of them directly trusting each other

nificant communication overhead, while its efficiency is inversely dependent on the network's latency and bandwidth.

- Error Rate and Fault Tolerance:
  - Error Correction Frequency: Both the schemes require CSS codes for quantum error correction. The frequency of such error correction processes, especially

in the case of the asymmetric scheme, impacts its performance.
  - Physical Qubit Quality: The efficiency of fault-tolerant operations depends on the error rates of physical qubits. The operations and qubits, being of high fidelity, the need for frequent error correction will be reduced, which in turn will lead to better overall efficiency.

- Ancillary Resource Management:
  - Symmetric Scheme: Since the T gates require ancillary states, their use requires their management, including pre-computation. The efficiency of the scheme is affected by how well these ancillary states are prepared and stored.
  - Asymmetric Scheme: The need for ancillary resources is not an issue because of its periodic interaction, but the management of random errors and their correction may introduce complexity.
- Scalability and Parallelism:
  - Scalability: The crucial feature will be how the schemes can scale with increasing problem size. The scalability of the encryption, decryption, and gate evaluation with respect to the number of qubits and gates can affect overall performance.
  - Parallelism: The degree to which operations can be parallelized is another factor. Schemes that can incorporate more parallelism in gate evaluations or error corrections can have a good performance.
- Resource Overhead:
  - Qubit Overhead: The ratio of logical to physical qubits that are required. If overhead is very high, the amount of logical qubits that can be computed at the same time decreases significantly, affecting efficiency.
  - Gate Overhead: This is the measure of physical gates that must be used to implement logical gates. More complex fault-tolerant implementations may increase gate overhead and hence affect overall efficiency.
- Implementation Complexity:
  - Algorithmic Complexity: Complexity of the key generation, encryption, decryption, and gate evaluation algorithms.
  - Hardware Requirements: Specific quantum hardware requirements, such as the need for specific types of gates, error-correcting codes, or interaction protocols.Scheme may require specialized quantum hardware with certain gate sets or interaction protocols, which will influence the practicality and efficiency of implementation.

Table 4 compares the efficiency of symmetric and asymmetric methods, focusing on key generation, encryption, decryption, and gate assessment. Symmetric schemes have no communication overhead, but fault-tolerant gate assessment necessitates extensive interaction due to ancillary states. Asymmetric schemes handle mistakes and communication through periodic interactions, resulting in lower error rates but increased overhead due to frequent re-encryption and interaction.

QOTP based scheme demonstrates a comprehensive approach to encryption, encompassing both classical and quantum operations.

Key generation efficiency: Key generation efficiency primarily depends on several factors:

- Matrix Operations: Key generation involves matrix operations, particularly generating the public key matrix A and the gadget matrix G.
- Randomness Generation: Generating randomness parameters, including r, y, and μ, also contributes to key generation time.
- Parameter Tuning: Choosing parameters such as modulus q, trapdoor bounds, and other cryptographic parameters also affects the efficiency of the key generation process.

Encryption efficiency: The efficiency of the encryption process is influenced by:

- Matrix Multiplication: Encryption involves matrix multiplication, mainly between the public key matrix A and the randomness matrix R.
- Gaussian Sampling: Sampling from Gaussian distributions, which forms the primary source of noise in the encryption process, adds computational overhead.
- Modular Arithmetic: Various operations on encrypted data occur with modular arithmetic due to the use of modulus q, which can affect the speed of encryption.

Decryption efficiency: Efficient decryption relies on:

- Matrix Operations: Decryption involves similar matrix operations as encryption, but one step further, wherein the original message needs to be recovered from the decrypted result.
- Modular Inversion: To recover the original message from the decrypted ciphertext, modular inversion is required.

Gate evaluation efficiency: Efficiency in gate evaluation is influenced by:

- Quantum Operations: In the quantum form of Fully Homomorphic Encryption, gate evaluations involve quan-

Table 4 Efficiency comparison between symmetric and asymmetric schemes with use case

Scheme	Property	Processing	Communication overhead	Use case
Symmetric	Key Generation Efficiency	Choosing and processing matrices from Q_n	No overhead at this stage	Suitable for secure data transfer within trusted internal networks due to minimal latency and efficient resource usage
	Encryption Efficiency	CSS encoding, converting each logical qubit into n physical qubits	No overhead at this point	Efficient for low-latency encryption in controlled environments
	Decryption Efficiency	CSS decoding of the ciphertext	No overhead at this point	Optimal for trusted systems where frequent decoding fidelity is required
	Gate Evaluation Efficiency	Fault-tolerant evaluation using transversal gates and ancillary states	Significant in the case of T gates due to the ancillary state	Ensures robust security while maintaining operational efficiency in trusted settings
Asymmetric	Key Generation Efficiency	Determination of private key and computation of $\hat{G}$	No overhead at this point	Ideal for distributed systems requiring periodic key generation across multiple participants
	Encryption Efficiency	CSS encoding with additional random errors	No overhead at this point	Suitable for cloud-based quantum platforms with untrusted participants
	Decryption Efficiency	Decryption and re-encryption at each step of evaluation	Significant due to the interaction	Enables secure operations in environments with high network variability
	Gate Evaluation Efficiency	Fault-tolerant evaluation with interaction at regular intervals	High with a number of interactions	Provides scalability and security in distributed architectures with error correction

tum operations like the application of quantum gates on qubits.

- Classical Operations: A QFHE scheme would need classical operations for the manipulation of classical information, especially when one goes beyond the evaluation of the Clifford family of gates, e.g., Toffoli.
- Bootstrapping Overhead: Bootstrapping, although critical for maintaining the security of the encryption during homomorphic operations, creates substantial computational overhead.

Table 5 analyzes the efficiency of key generation, encryption, decryption, and gate evaluation. Key generation is efficient with polynomial complexity for both public and private keys, although the latter may be computationally intensive due to vector generation. QOTP encryption is extremely efficient, whereas traditional encryption has polynomial complexity and potentially heavy procedures. Decryption is generally as efficient as encryption, as it uses reverse quantum gate or classical ciphertext processes. Gate evaluation is efficient, with Clifford gates being polynomial in complexity and Toffoli gates requiring little additional overhead. Overall, the examination focuses on the balance of efficiency and computing demands across several operations.

For evaluating the efficiency of a QFHE scheme, especially scheme based on QOTP, more factors must be added to get a more thorough analysis.

- Communication Overhead:
  - Error Correction: Like all communication systems, Scheme has to deal with errors that may be introduced in its transmission and processing. Error correction mechanisms, such as error-correcting codes or redundancy, may be needed to make the communication reliable.
  - Fault Tolerance: The ability of Scheme to handle robustness against any fault, be it hardware failure or transient error, is important to maintain the integrity of the system. The scheme may make use of fault-tolerant mechanisms to handle the effects of such problems.The possible communication overhead will stem in the need for ancillary states for the T-gates.

Table 5 Efficiency analysis of key generation, encryption, decryption, and gate evaluation

Process	Efficiency analysis
Key Generation	Public Key Generation: Efficient; depends only on the polynomial complexity of matrix generation, usually $O(n^2)$ where n is the security parameter
	Private Key Generation: Polynomial in complexity, but vector generation through discrete Gaussian could be computationally heavy. Proper parameter selection is essential to strike a balance between security and efficiency
Encryption	Quantum One-Time Pad (QOTP) Encryption: Very efficient; only involves simple quantum operations like XOR gates
	Classical Encryption of Secret Pad: Polynomial in complexity, but may have heavy classical cryptographic operations like modular arithmetic
Decryption	Reverse Operations: Generally as efficient as encryption; reverse application of quantum gates or decryption of classical ciphertext using keys
Gate Evaluation	Clifford Gates: Efficient and polynomial in complexity. Toffoli Gate: Only involves classical operations on the key; hence efficient even if there is some overhead

How often and how much data is passed between Alice and Bob determines the overall efficiency.

- Ancillary Resource Management:
  - Memory Management: Efficient use of memory resources is aimed at reducing overhead and improving performance. Scheme should implement dynamic memory allocation, garbage collection, and efficient data structures to handle ancillary resources properly.
  - Energy Consumption: Energy consumption needs to be managed effectively, especially in a resource-constrained environment like mobile or IoT devices. Scheme may optimize cryptographic operations and resource utilization to minimize energy overhead.
- Scalability and Parallelism:
  - Scalability: Scheme should scale according to the data size and computational demands for large applications. Scalable algorithms and parallel processing techniques will offer efficient resource utilization and accommodate increasing workloads.
  - Parallelism: The parallel processing capability could be used in getting the best performance from Scheme by running different cryptographic operations simultaneously, thus saving time for processing.
- Resource Overhead:
  - Memory Overhead: Memory overhead in Scheme 21 is due to the storage requirements of cryptographic keys, intermediate data structures, and computation results. This would be minimized via efficient data representation and management.
  - Computationa Overhead: In the scheme, cryptographic operations cause computational overhead, namely modular arithmetic, matrix operations, and random number generation. Mitigation of the resource overhead is achieved by optimizing algorithms and data structures, which would thus reduce the computational complexity.
- Implementation Complexity:
  - Algorithmic Complexity: Cryptographic algorithms used within the scheme shall vary in their level of complexity. This would affect the ease of the implementation process. Simplification of implementation and maintenance efforts will take place by the clear documentation of the scheme's operation, standardized protocol, and modular design practices.
  - Hardware Requirements: Specific quantum hardware requirements, such as the need for specific types of gates, error-correcting codes, or interaction protocols.

Practical considerations

Quantum fault-tolerant based scheme: Here are some practical considerations specific to this scheme, focusing on fault-tolerant implementations and limitations on circuit complexity:

Symmetric Scheme:

Fault-Tolerant Implementation: This Scheme leverages fault-tolerant quantum computation for operations on encrypted data. This ensures robustness against errors during computations but requires additional resources for fault tolerance protocols.

Circuit Complexity Limitations: While the scheme supports universal quantum computations, deeper circuits with many T gates can be problematic. Ancillary states provided by Alice are essential for T gates, and their number scales with the number of T gates. This can become a bottleneck for complex computations.

Interaction-Free: The symmetric scheme operates without any interaction between Alice (who holds the secret key) and Bob (who performs computations). This simplifies the communication overhead but requires Alice to anticipate the number of T gates in advance to provide enough ancillary states.

Asymmetric Scheme:

Interaction Requirement: Unlike the symmetric scheme, the asymmetric scheme introduces periodic interaction between Alice and Bob. After each computation step, Bob sends the intermediate result to Alice for decryption and re-encryption. This interaction ensures error control but adds communication overhead and potential latency.

Security based on Public-Key Cryptosystem: The security of the asymmetric scheme relies on a public-key cryptosystem similar to Fujita's scheme. The formal security proof is not provided in the paper but is mentioned for future work.

Error Correction and Fault Tolerance: Similar to the symmetric scheme, fault-tolerant quantum computation is employed for operations on encrypted data. However, unlike the symmetric scheme, the asymmetric scheme doesn't involve explicit periodic error correction procedures.

Additional Considerations:

Quantum Bootstrapping: The paper mentions quantum bootstrapping as a potential solution to avoid interaction in the asymmetric scheme. This is an active research area, and its feasibility for Scheme 18 remains to be explored.

Commutative Properties: The paper highlights the importance of the encoder (CSS code) commuting with most quantum operators (H and CNOT gates). The T gate is the exception, necessitating ancillary states or interaction for its implementation.

QOTP based scheme: Here are some practical considerations specific to this scheme:

Ensuring Circuit Privacy and Noise Rate: This scheme adheres to the circuit privacy argument put forth by [42] and relies on the GSW encryption scheme, which utilizes a polynomial noise rate. Therefore, it is crucial to properly control the noise during encryption to prevent any information leakage.

Choosing the Modulus: The choice of modulus (q) holds significant importance in this scheme, even though the hardness assumption does not depend on it. The modulus selection impacts both the efficiency and security of the encryption scheme, affecting the size of the ciphertext and computational complexity.

Propagating Randomness: The propagation of randomness introduces Gaussian noise into the implementation. Properly managing Gaussian noise is indeed vital in maintaining security while minimizing computational overhead.

QFHE Implementation: This scheme proposes the utilization of QFHE. Quantum computing introduces additional complexities, such as qubit error rates, gate fidelities, and decoherence, which must be addressed in order to ensure the scheme's effectiveness.

Key Encapsulation Mechanism (KEM): This scheme is premised on a KEM for QOTP encryption. The security of the KEM is paramount, and great care must be taken regarding its analysis, which might involve further complications due to additional steps like post-quantum security to future-proof against quantum attacks.

CNOT Evaluation: The capacity to evaluate CNOT gates classically is the heart of quantum computation. Executing classical operations on quantum-encrypted data needs efficient algorithms and proper management of noise and error rates.

Parameter Selection: The selection of parameters such as q, σ, and p needs to be appropriate. Their oversight could lead to vulnerabilities or inefficient performance of the encryption scheme.

Circular Security Assumption: The scheme assumes circular security, which might increase the complexity of the security proof and, possibly, may require additional validation for being robust against potential attacks.

Bootstrapping:This module enables the scheme to support deeper computations but introduces computational overhead and complexity into the scheme. Efficient bootstrapping techniques are crucial for practical implementations of the scheme.

Lattice Security: The security proofs in the scheme are based on the hardness of lattice problems. Ensuring the hardness of these problems, being under a variety of attacks and assumptions, is crucial for the overall security of the scheme.

Comparison of QFHE with classical FHE

Computational Efficiency: The quantum fault-tolerant-based QFHE scheme leverages classical error-correction techniques, particularly CSS codes, to ensure that quantum operations remain efficient. By employing error correction, this scheme minimizes the overhead associated with maintaining quantum coherence during computations. This leads to greater computational efficiency, especially for large-scale encrypted computations, as it can process more data without the exponential resource costs associated with classical encryption schemes. On the other hand, the QOTP-based scheme combines the concept of a QOTP with classical

FHE. While this combination provides a quantum layer of encryption, it requires performing both quantum and classical operations, which introduces additional computational overhead. Thus, this scheme is less efficient than the quantum fault-tolerant scheme in terms of raw computational speed. Classical FHE, however, is the least computationally efficient due to its reliance on polynomial-time operations, which grow exponentially with data size, resulting in significant overhead for encryption and decryption processes.

Noise Management: Noise management is a critical challenge in quantum computations. The quantum fault-tolerant QFHE scheme addresses this issue by utilizing classical error-correction codes like CSS codes, which significantly reduce the impact of noise during quantum operations. While this method improves the reliability of computations, the level of noise that can be effectively managed is still limited by the efficiency of the error-correction codes and the scale of the system. This means that although the quantum fault-tolerant QFHE scheme performs well in maintaining data integrity, noise still poses a potential problem, especially as the system scales. The QOTP-based scheme also benefits from the inherent noise resistance of QOTP encryption, which protects the data from quantum noise during encryption. However, once combined with classical FHE, noise can accumulate during the classical processing phase, potentially degrading the overall reliability. Classical FHE does not offer inherent noise resistance and faces the challenge of noise propagation through successive encryption and decryption layers. The noise in classical FHE increases with every operation, and noise management typically requires computationally expensive bootstrapping techniques.

Security: Both QFHE schemes provide enhanced security features over classical FHE. The quantum fault-tolerant scheme, through its error correction methods, provides a more robust defense against potential quantum attacks, ensuring that even if parts of the quantum system are compromised, the integrity of the encrypted data remains intact. Similarly, the QOTP-based QFHE scheme benefits from the QOTP's security, which provides theoretically perfect encryption, given that the quantum key is never reused. However, the combination with classical FHE can introduce vulnerabilities related to the classical layers of encryption. Classical FHE, while secure in classical computing environments, is inherently vulnerable to the increasing capabilities of quantum computers, which may be able to break classical encryption methods in the near future.

Scalability: Scalability is another important factor in comparing QFHE and classical FHE schemes. The quantum fault-tolerant QFHE scheme is highly scalable due to the use of efficient error-correction codes and quantum parallelism, allowing it to handle large volumes of data more effectively. This scalability is a major advantage over classical FHE, which suffers from exponential growth in computational complexity as the size of the encrypted data increases. Although the QOTP-based QFHE scheme is scalable in terms of quantum encryption, the additional classical FHE layer makes it somewhat less scalable, especially as the system size grows. The need to manage both quantum and classical operations introduces a level of complexity that can limit scalability compared to the more streamlined quantum fault-tolerant scheme.

Implementation Complexity: The implementation of the quantum fault-tolerant QFHE scheme requires advanced quantum error-correction techniques, making it more complex to implement than classical FHE. The use of CSS codes and the need to maintain quantum coherence through error-correction protocols add additional layers of complexity to the system. Similarly, the QOTP-based QFHE scheme, while more secure, requires the combination of quantum and classical encryption mechanisms, adding implementation complexity. In contrast, classical FHE, although computationally expensive, is relatively simpler to implement in classical computing environments, which may make it more accessible in the short term but less resilient against quantum threats.

Overhead and Resource Consumption: The overhead and resource consumption of QFHE schemes are influenced by both quantum and classical elements. The quantum fault-tolerant QFHE scheme, with its reliance on quantum error correction, has moderate overhead in terms of quantum resources, but it is more efficient than the QOTP-based scheme in terms of resource consumption. The QOTP-based scheme requires both quantum resources for the one-time pad encryption and classical resources for the homomorphic encryption operations, leading to higher overall resource consumption compared to the quantum fault-tolerant scheme. Classical FHE, while requiring significant computational resources, generally consumes fewer quantum resources as it operates in a purely classical environment.

Application

In this section, we illustrate the multi-faceted applications of QFHE and its associated schemes and describe the profound impact of QFHE on various domains of secure computation.

Secure Quantum Voting Protocols: QFHE enables the development of secure quantum voting protocols where voters can cast their encrypted ballots without disclosing their preferences. The principles of QFHE ensure that every vote remains confidential during the whole process of voting, thereby prohibiting manipulation or leakage of the votes. This work builds on the ability of QFHE to perform computations on encrypted data, enabling the tallying of votes while preserving the privacy of the voters [43].

Practical Implementations in Quantum Systems: The paper also discusses practical applications of QFHE in bosonic systems and quantum optics, with implications for quantum communication and computation systems. QFHE provides information-theoretic security based on quantum mechanics, which has major implications for privacy-preserving data processing and safe quantum computing applications [44].

Privacy-Preserving Quantum Computations Using QFHE: QFHE is applied to achieve secure quantum computations by leveraging the unique properties of quantum states. In the application under discussion, QFHE performs computations on encrypted data using single-photon states encoded with information in their polarization. This technique enables secure processing of quantum data without disclosing the underlying information, resulting in privacy-preserving delegated quantum computing. QFHE's general applicability includes aiding secure operations on sensitive quantum data, which is critical in scenarios where data privacy and security are vital, such as secure cloud-based quantum computing and sensitive data analysis [45].

Enhanced Multi-Party Computation with QFHE: The improved QFHE scheme proposed in [46] extends the flexibility of QHE by enabling computations involving all single-qubit unitary operations, even when the quantum capability of evaluators is limited. This advancement is especially valuable in collaborative settings where numerous evaluators collaborate to execute calculations on encrypted data, solving trust and network load issues that arise with single-evaluator systems. This enhanced approach illustrates a realistic application of QFHE in safe, multi-party calculations by lowering the quantum requirements and showing viability via simulations on IBM's cloud quantum computing platform.

Limitations

Quantum-based encryption, while promising for secure communication and computation, faces several notable limitations that hinder its widespread adoption. One significant challenge lies in the sensitivity of quantum systems to noise and decoherence, which can compromise the stability and reliability of encrypted data during computation or transmission. Additionally, quantum encryption protocols often require advanced hardware and precise control of quantum states, making them resource-intensive and difficult to scale for practical applications. Key management in quantum systems also presents unique challenges, as maintaining synchronization and preventing vulnerabilities such as eavesdropping during key distribution can be complex. Furthermore, the high computational overhead and lack of standardization across quantum platforms add to the difficulties of implementing quantum encryption on a broader scale. Addressing these limitations is essential to unlock the full potential of quantum-based encryption technologies.

In addition to the conventional challenges faced by quantum cryptographic systems, emerging cyber threats pose significant risks to their effectiveness. While quantum cryptography, particularly Quantum Key Distribution (QKD), promises enhanced security through principles of quantum mechanics, it remains vulnerable to sophisticated cyberattacks. For example, state-sponsored attackers and advanced persistent threats (APTs) exploit weaknesses in quantum hardware and software, making even theoretically secure systems susceptible to compromise. Moreover, AI-driven cyberattacks have the potential to bypass existing quantum safeguards, further threatening the robustness of these systems in real-world scenarios. The study [47] provides a comprehensive analysis of these vulnerabilities, highlighting how new attack vectors-ranging from hardware flaws to AI-assisted side-channel attacks-can undermine the security of quantum cryptography systems. This perspective underscores the need for ongoing research to identify and mitigate these evolving threats, ensuring that quantum cryptographic protocols can withstand future challenges posed by increasingly sophisticated cyber adversaries.

Research issues in QFHE

The development of QFHE schemes is a growing field of study aiming at providing secure processing on encrypted quantum data. This subsection explores several QFHE schemes and their respective challenges, with a focus on applying conventional FHE concepts to quantum settings, providing effective security measures, and increasing computational performance in the face of quantum-specific constraints. Each strategy addresses distinct issues such as key management, encryption fidelity, and multi-party computation, reflecting broader research objectives of improving QFHE's applicability and dependability in quantum computing contexts.

The development of QFHE libraries is an important research topic in quantum computing. These libraries attempt to provide accessible, dependable implementations of fully homomorphic encryption methods designed specifically for quantum circuits. Unlike classical FHE libraries, which have experienced extensive research and implementation, QFHE libraries are still in their early phases. The challenges include adapting conventional FHE algorithms to quantum settings, ensuring interoperability with quantum computing platforms, and resolving quantum circuits' unique security and performance considerations. Robust QFHE libraries are critical for advancing practical applications in secure quantum computation, needing continued investigation and development of both theoretical foundations and practical implementations.

The paper [48] describes a new QFHE system that employs separate keys for encryption and decryption, distinguishing it from prior schemes that used the same key for both processes. This technique, based on the TBQC model, incorporates interactive computation into the evaluation process and provides perfect security via QOTP in encryption and decryption. It is ideal for safe delegated quantum computing between two parties, allowing the server to compute encrypted data without knowing the decryption key. However, obstacles include the need for interactive computation and secure key distribution. General research difficulties in the QFHE sector resulting from these issues include improving key management protocols, increasing efficiency and scalability, and establishing viable ways for secure multiparty quantum computing.

The scheme [38] aims to ensure that ciphertexts in QFHE retain the features required for secure and efficient computation. Various QFHE techniques utilize superpositions of states, randomization processes, and the use of lattice and Gaussian distribution features to produce desired cryptographic outcomes. However, these approaches frequently encounter limits in terms of trace distance, uniform distribution, and high probability decryption to the correct value. Specific issues include regulating the complexity of the Fourier Transform steps, limiting norm differences, and making optimum use of lattice features. General research questions raised by these obstacles include improving the robustness of randomization algorithms, increasing the efficiency of Fourier Transform implementations, and improved handling of Gaussian distributions inside lattice-based cryptography frameworks. These findings point to larger areas of improvement required to advance the field of QFHE.

The suggested QFHE techniques in [49] concentrate on quantum obfuscation using multi-valued quantum point obfuscation, which covers both single-qubit and multi-qubit cases. These approaches, independent of secret keys, ensure that the mean values of encryption and evaluation outputs are mixed, preventing attackers from obtaining information without prior knowledge of plaintext distribution, hence improving security. Despite overcoming low-dimension restrictions, obstacles remain in secure multiparty computation, highlighting broad QFHE research issues such as enhancing security measures, dealing with higher-dimensional quantum states, and ensuring efficient computation in a quantum environment. These areas illustrate greater hurdles to expanding QFHE technology.

The QFHE approach reported in [50] allows for secure assessment of quantum circuits on encrypted qubits without prior knowledge of the circuit form. Features include the use of auxiliary qubits and classical bits for encryption, which ensures that operations adhere to classical homomorphic cryptosystem principles. Limitations include imperfect visibility in two-qubit gates, which leads to decryption problems, vulnerability to attacks leveraging detection events and key management flaws, and practical limits such as low detection rates, which affect overall security. Challenges highlighted across schemes include increasing gate fidelity, strengthening key management against eavesdropping, and scaling up for realistic quantum computing applications. These concerns highlight ongoing research efforts to provide more efficient, safe, and scalable QFHE protocols suitable for larger quantum computing deployments.

Threat model for QFHE systems

A threat model identifies potential risks and attack methods that could compromise the security of a system. In the case of QFHE, the main goal is to protect sensitive data while allowing computations on that encrypted data. Below is a simplified breakdown of possible attack vectors in a QFHE system:

1. Quantum Attacks on Encryption Keys

Quantum computers have the potential to break traditional encryption methods, such as RSA or ECC, which rely on the difficulty of factoring large numbers or solving elliptic curve problems. If a quantum adversary gains access to encryption keys, they could potentially decrypt sensitive data, even if it was encrypted using QFHE. However, the QFHE scheme itself is designed to withstand quantum attacks, but this doesn't eliminate the risk of key exposure.

2. Side-Channel Attacks

Side-channel attacks exploit unintended information leaked by the system during computations. For example, an attacker could observe the physical behavior of the quantum device, such as timing or power consumption, to infer sensitive information like encryption keys or intermediate computation results. These attacks are more common in classical systems but are still a risk in quantum environments, especially if the hardware isn't secure.

3. Quantum Noise and Errors

Quantum systems are susceptible to quantum noise and errors due to the fragile nature of quantum states. If an attacker can manipulate the quantum environment (e.g., introducing errors), they might distort the results of computations or decrypt data. Although QFHE systems include error correction, an adversary who controls the quantum environment might still find ways to exploit these errors.

4. Man-in-the-Middle (MITM) Attacks

In a Man-in-the-Middle attack, an attacker intercepts and possibly alters the communication between two parties (e.g., Alice and Bob in QFHE). If the quantum communication is not properly authenticated or if the encryption scheme is

weak, an attacker could decrypt or manipulate the exchanged quantum information, compromising the security of the system.

5. Compromised Trusted Parties

QFHE often involves two or more trusted parties (e.g., Alice and Bob). If either of these parties is compromised, the security of the entire system is at risk. For example, if Bob's private key or quantum device is hacked, they could potentially decrypt encrypted data or manipulate the computations.

Future directions

As quantum technologies continue to evolve, the field of QFHE presents numerous opportunities for further exploration and development. One promising direction is the design of more efficient QFHE schemes with reduced computational overhead and improved scalability. Current schemes often face challenges in terms of error correction and resource optimization, which must be addressed to make QFHE practical for large-scale implementations.

Another critical area for future research is the development of hybrid cryptographic frameworks that combine classical and quantum techniques. Such frameworks could leverage the strengths of both approaches, providing robust security while minimizing the limitations associated with fully quantum systems. In particular, exploring chaos-based or other unconventional encryption methods within a quantum context could yield innovative solutions for secure data processing.

Furthermore, QFHE has significant potential in quantum cloud computing environments. Future studies could investigate methods to enhance the interoperability of QFHE with quantum cloud platforms, enabling seamless and secure encrypted computations for sensitive applications like financial transactions, healthcare data processing, and government communications.

Addressing the cost barriers of quantum computing is another essential focus. Research into cost-efficient hardware, error-tolerant architectures, and accessible software tools will be vital for making QFHE technologies more practical and widespread. Additionally, collaboration between academia and industry could accelerate the development of real-world applications, bridging the gap between theoretical advancements and practical deployment.

Finally, ongoing efforts should prioritize rigorous security analysis to address potential vulnerabilities and ensure resilience against both classical and quantum threats. As quantum computing matures, the cryptographic landscape will continue to evolve, requiring proactive innovation to stay ahead of emerging challenges. These future directions underscore the importance of sustained research and interdisciplinary collaboration to unlock the full potential of QFHE in securing data in the quantum era.

Hierarchical memory management with quantum caching and reinforcement learning for QFHE schemes

While existing QFHE schemes such as the two schemes discussed in above section provide useful functionalities, they tend to be memory-intensive: this is because of the requirements of fault-tolerant quantum computation, the circuit complexity, the management of noise, and the details of QFHE implementation. The survey revealed a common challenge across various QFHE schemes: the need for efficient memory management to handle complex computations without exceeding hardware constraints. To address these limitations and enhance the overall efficiency of QFHE computations, we propose a novel hierarchical memory management system.This paper presents a new hierarchical QFHE memory management system, utilizing quantum cache and reinforcement learning algorithms for optimal dynamic data movement. This method is to reduce the quantum memory footprint but also be efficient.Memory optimization strategies, such as the hierarchical memory management system, may aid in memory intensity reduction by dynamically managing memory resources in response to access patterns and computational needs.

The reinforcement learning agent optimizes memory usage by selecting and retaining the most relevant data for improving future decisions, based on feedback signals (reinforcements) received from its interactions with the environment [51]. Specifically, the RL agent monitors access patterns and operational needs of the QFHE system and predicts the most efficient memory allocation strategies. Through a process of trial and error, the agent learns which data should be kept in quantum memory (cache) and when it should be moved to classical memory or offloaded. This continuous feedback loop allows the system to adapt to varying workloads and quantum operations.

In technical terms, the RL agent uses a policy gradient method or Q-learning to evaluate memory allocation decisions. At each step of computation, the agent receives a "reward" based on the efficiency of memory usage-measured in terms of computational time, memory load, or error rates-and adjusts its policy to maximize this reward. The reward function provides feedback from the environment, guiding the agent to maximize long-term rewards by associating specific rewards with states or state-action pairs, while the policy function defines the mapping from states to actions, which can be either stored or dynamically computed based on environmental interactions [52]. By constantly refining its memory management strategy, the agent ensures that the

QFHE system operates within the hardware's memory constraints, while minimizing delays or resource exhaustion.

Memory optimization strategies, such as the hierarchical memory management system, may aid in memory intensity reduction by dynamically managing memory resources in response to access patterns and computational needs. This approach not only addresses memory limitations but also improves the overall efficiency of QFHE systems, making them more viable for real-world applications.

System components

Quantum Registers: A small portion of the quantum registers is dedicated as a high-speed cache, called Q-Cache. In Q-Cache, commonly accessed qubits are stored along with some crucial entangled qubit pairs, which would be necessary for computational purposes.

Classical Memory with Access History: The rest of the qubits and the related classical data are placed in classical memory, namely C-Memory. The classical memory maintains an access history for each qubit and data element. It tracks the frequency and timestamps of access.

Reinforcement Learning Agent: A lightweight RL agent is tasked to watch over the QFHE computations. It observes the access patterns of qubits and data elements that are stored in C-Memory. The analyzed parameters include the following:

- *Access Frequency:* The number of times a certain qubit or entangled pair is accessed while conducting the computation.
- *Temporal Locality:* Whether multiple accesses to the same qubit or entangled pair happen in a time window.
- *Spatial Locality:* Whether the accessed qubits or entangled pairs that are accessed frequently are logically related to each other-for example, does the same quantum circuit have it.

Data Transfer Protocol: There is a need for a communication protocol to support the movement of qubits and entangled pairs between the Q-Cache and the C-Memory as per the observations and recommendations of the RL agent. The data transfer protocol should be optimized to ensure minimal overhead and error correction in the data movement.

Figure 3 outlines a hierarchical memory management system designed to optimize computations in QFHE. The diagram depicts the integration of a Q-Cache, which stores frequently accessed data to minimize retrieval times, and a Reinforcement Learning (RL) Agent, which dynamically manages the movement of data between different levels of memory. This approach aims to enhance computational efficiency by ensuring that critical data is readily available while adapting to changing computational needs. By optimizing data storage and retrieval, this system can significantly improve the performance of QFHE operations, making it more practical for complex and resource-intensive tasks.

Algorithm

The following presents an algorithm for Hierarchical Memory Management using RL specifically designed for QFHE computations. This innovative approach seeks to optimize the movement of data between classical memory (Cmem) and a dedicated quantum cache (Qcache) to enhance the overall efficiency and performance of QFHE operations.

The algorithm operates as follows: Initially, an RL agent is tasked with monitoring the access patterns of qubits within the Cmem. By analyzing historical data, including frequency of access and both temporal and spatial locality, the RL agent gains insights into the usage patterns of qubits. This information is crucial as it helps the agent understand which qubits are frequently accessed and which ones are less utilized.

Based on this analysis, the RL agent makes informed recommendations for data management. For instance, it might suggest caching frequently accessed qubits in the Qcache to ensure faster access and reduce retrieval times. Conversely, it might recommend swapping out less utilized qubits from the Qcache to make room for more critical data. The recommendations are then implemented using sophisticated error correction protocols, which ensure that data integrity is maintained during the movement process.

After the recommendations are executed, the RL agent updates the access history to reflect the latest data movement. This updated history serves as the basis for future decisions, allowing the RL agent to continually refine and optimize its policy. The RL agent's learning process is iterative: it adjusts its strategies based on the observed impact of its previous decisions on QFHE efficiency.

Over time, this dynamic and adaptive approach enables the hierarchical memory management system to significantly improve the performance of QFHE computations. By efficiently managing data movement and optimizing memory usage, the algorithm enhances computational efficiency, making it feasible to handle more complex and resource-intensive QFHE tasks with greater effectiveness.

Theoretical proof

The Hierarchical Memory Management method for proposed QFHE schemes involves the use of an RL agent to guide dynamic data movement between quantum cache (Q-Cache) and classical memory (C-Memory), taking into account access patterns and efficiency metrics. Although a strict mathematical proof for this is difficult because of the complexity and variability of real-life situations, we can now

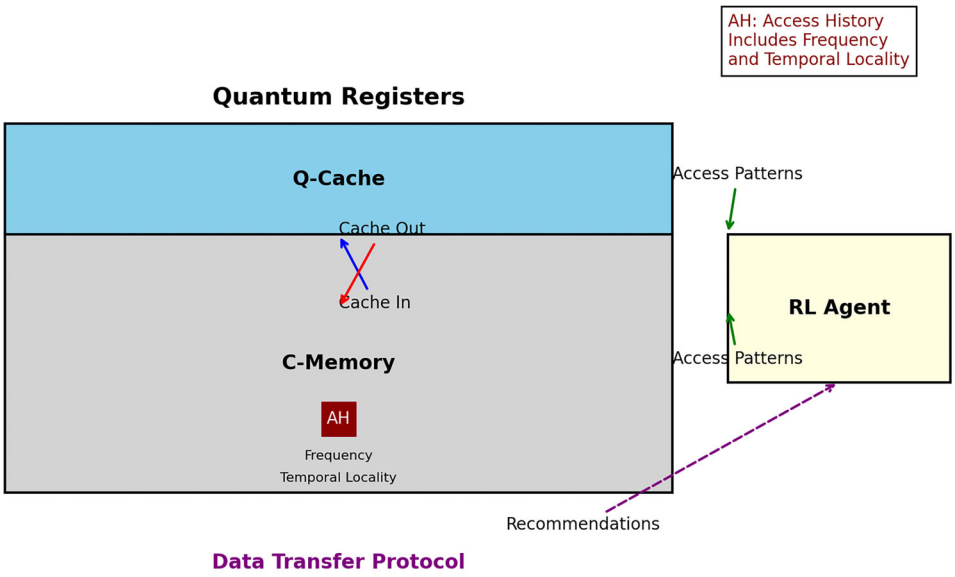

Fig. 3 This diagram outlines a hierarchical memory management system for optimizing QFHE computations. It leverages a Q-Cache for frequently accessed data and an RL Agent to dynamically manage data movement

outline a framework theoretically, asserting the effectiveness of the approach at every point:

MDP Formulation: The problem of memory management can be cast as an MDP, where the RL agent interacts with the environment (QFHE computations) towards learning optimal data movement policies. The state space consists of different configurations of data in both the Q-Cache and C-Memory, while the action represents how to move qubits between these tiers.

Reward Function Design: The reward function defines the criteria through which the RL agent receives feedback on the quality of its decisions. Building an appropriate reward function that rewards the optimization of QFHE computations will urge the agent to learn policies that will result in better performance.

Optimality Analysis: Theoretical analysis can focus on showing the convergence of learning processes of the RL agent toward the optimum memory management policies. This may involve showing conditions under which the policies of the RL agent will lead to the optimal solution, thus the efficient use of resources over time.

Complexity Analysis: Theoretical investigation into the computational complexity of the proposed method, including considerations such as time complexity of decision algorithms, memory overhead for maintaining access history, and convergence properties of the RL algorithm.

Performance Guarantees: This includes theoretical guarantees of the proposed memory management system's performance under changing conditions in the workload characteristics, hardware constraints, and error correction mechanisms that will be present, since these are guaranteed to make the method robust and reliable in practical settings.

While a complete theoretical proof may not be feasible-since the QFHE schemes and real-world environments are inherently complex and variable-the theoretical framework outlined goes a long way to further the analysis and understanding of the effectiveness of the proposed method. Empirical validation in the forms of simulations and experiments will complement the theoretical insights by demonstrating the practical utility and benefits derived from the hierarchical memory management approach in the enhancement of the efficiency in QFHE computations.

Analytical model to estimate the memory footprint reduction and efficiency of the method.

Using this model, we can determine the potential advantages offered by the new proposed hierarchical memory management system in terms of memory footprint reduction and efficiency improvements.

Factors

Total Number of Qubits (Q_{total}): Total number of qubits in the quantum processing unit (QPU).

Cache Size (Q_{cache}): A portion of the total qubits Q_{total} dedicated for Q-Cache.

Access Probability Distribution (P_{access}): The probability distribution of a given qubit/entangled pair being accessed by computation in QFHE. This can be derived from historical access information or known access patterns for a particular QFHE scheme.

Cache Hit Rate (H_{rate}): The probability that the required qubit/entangled pair is found in the Q-Cache during compu-

Input: Q_{total}: Total number of qubits in quantum registers,
C_{mem}: Classical memory used for storing qubits and access history,
Q_{cache}: Designated portion of Q_{total} allocated for quantum cache,
RL_{agent}: Reinforcement learning agent equipped with a reward function,
$access\_history$: Data structure within C_{mem} recording access history for each qubit or data element.
Output: Optimized data movement strategy between Q_{cache} and C_{mem} to enhance efficiency of QFHE computations.
Initialization:

- Initialize RL_{agent} with a reward function designed to maximize efficiency, minimize cache misses, and reduce the number of data swaps.
- Populate $access\_history$ with initial access counts for each qubit or data element stored in C_{mem}.

while *QFHE computation is ongoing* **do**
RL_{agent} monitors access patterns of qubits in C_{mem}.
Analyze $access\_history$ to determine:

- Access frequency of each qubit or entangled pair.
- Temporal locality (recent access patterns).
- Spatial locality (logical relationships between accessed qubits).

Based on the analysis, RL_{agent} recommends data movement:

- *Cache In:* Move a qubit or pair from C_{mem} to Q_{cache} if it shows high access frequency or strong locality.
- *Cache Out:* To maintain space in Q_{cache}, move less frequently used qubits from Q_{cache} to C_{mem}.

Execute the data transfer protocol as recommended by RL_{agent}:

- Transfer qubits or pairs between Q_{cache} and C_{mem} with necessary error correction.
- Update $access\_history$ in C_{mem} to reflect the data transfers.

RL_{agent} evaluates the impact of its decisions on QFHE efficiency and updates its policy according to the reward function.
end

Algorithm 1: Hierarchical memory management with RL for QFHE

tation. It depends directly on the efficiency of learning by the RL agent and its decision-making capabilities.

Data Transfer Overhead ($T_{overhead}$): Cost (in terms of time/energy) to transfer a qubit/entangled pair between Q-Cache and C-Memory.

Assumptions

- The access probability distribution (P_{access}) of qubits and entangled pairs is either known or can be estimated with a high degree of assurance for the selected QFHE scheme.
- The learning capacity of the RL agent, along with data movement decisions that lead to the cache hit rate being high, or in other words, the RL agent is able to make the right decision.
- Overhead to transfer data (T_{overhead}) is invariant for all qubits and entangled pairs.

Metrics to estimate

Memory Footprint Reduction ($M_{reduction}$): Percentage reduction in memory usage on the QPU, with and without a hierarchical memory management system.

Computation Time Improvement ($T_{improvement}$): Percentage improvement in total computation time taken, because of data movement being effective and efficient due to the RL agent.

Model equations

Average Cache Misses per Computation (M_{miss}): This calculates the number of times, on average, the RL agent requires data swapping between Q-Cache and C-Memory in a single QFHE computation.

$$M_{\text{miss}} = Q_{\text{total}} \times (1 - H_{\text{rate}})$$

Memory footprint reduction ($M_{reduction}$): This computes the percentage of QPU memory that the hierarchical system saves.

$$M_{\text{reduction}} = \frac{M_{\text{miss}} \times T_{\text{overhead}}}{Q_{\text{total}} \times T_{\text{computation}}} \times 100\%$$

Where, $T_{\text{computation}}$: Average computation time of the QFHE scheme without the memory management system.

Computation time improvement ($T_{improvement}$): It estimates the potential computation time that can be improved due to reducing data transfer overhead.

$$T_{\text{improvement}} = \frac{M_{\text{miss}} \times T_{\text{overhead}}}{T_{\text{computation}}} \times 100\%$$

Sensitivity analysis

We take a sensitivity analysis to see how model estimates vary with different assumptions. This shows how changes in such key parameters as cache hit rate, H_{rate}, and data transfer overhead, T_{overhead}, affect the predicted memory footprint reduction and computation time improvement. We consider H_{rate} ranging between 0.5 and 0.9 and vary T_{overhead} between 1 and 10 to look at the resulting changes in $M_{\text{reduction}}$ and $T_{\text{improvement}}$. This enables one to see how robust the method proposed here is under different conditions and indicates critical factors that influence its performance.

Incorporation of specific QFHE schemes

Generalization for specific QFHE schemes can be achieved by further developing the model to incorporate already known access patterns related to QFHE schemes into the P_{access} distribution. For example, BGV and CKKS have known access patterns since they have different structures according to their mathematical principles and operational requirements. The access patterns can be incorporated into the model, which would then be able to predict the memory reduction and efficiency gains under different scenarios in a more tailored manner. This can be done through the analysis of the specific access frequency and locality characteristics of the chosen QFHE scheme and adjusting the P_{access} distribution accordingly.

Limitations and future refinement

This model is based on the assumptions of access patterns and effectiveness of the RL agent. The quality of these assumptions and the QFHE scheme selected highly affect the quality of the estimates. To enhance the model, it can be further tailored to specific QFHE schemes by integrating the known access patterns of those schemes into the P_{access} distribution. For example, this could involve analyzing how often particular qubits or quantum gates are accessed during computations and incorporating this data into the model.

Additionally, more advanced versions of the model could account for the fact that different qubits or entangled pairs may experience varying levels of data transfer overhead due to factors such as physical distance, quantum decoherence, or the specific error-correction mechanisms in use. By refining these aspects, the model could provide more precise estimates and better reflect the practical constraints and behaviors of real-world QFHE systems.

Conclusion

In this paper, we have carried out a comprehensive survey for two of the key QFHE schemes in terms of their foundational principles, mathematical frameworks, security considerations, and practical implications. From the survey we can understand that the schemes are memory-intensive in nature and need to be so, since the scheme demands fault-tolerant quantum computation, circuit complexity, and noise management. In this direction, we proposed a novel hierarchical memory management system that leverages quantum caching and reinforcement learning to optimize the movement of data and reduce the footprint of memory. The proposed system dynamically manages frequently accessed qubits and entangled pairs by storing them in a high-speed quantum cache, called Q-Cache. In the lightweight reinforcement learning architecture, the QFS RLA monitors and adapts to access patterns. Based on the access pattern, the RL agent's recommendation for data movement minimizes cache misses and data transfer overhead, thus enhancing the overall efficiency of computations using QFHE. We developed an analytical model to estimate the potential benefits of our proposed technique for the considered access probabilities, hit ratios of the cache, and overhead on data transfer. It provides the analytical metrics of reduction in memory footprints and improvement in computation time to illustrate the effectiveness of our method in using resources with QFHE schemes. Our theoretical framework illustrates that the potential of a hierarchical memory management system is to dramatically increase the efficiency of QFHE computations. Nevertheless, the empirical validation by simulations and experiments will have to be established to confirm the practical utility and benefits of the theory. In conclusion, the proposed method implies an advance in the field of quantum secure computation. By considering the problem of memory management in QFHE schemes, we pave a way to efficient and scalable quantum computation schemes, giving impetus to further development and applications in this exciting area.

References

1. Malik DS, Mordeson JN, Sen M (1997) Fundamentals of abstract algebra. McGraw-Hill, New York
2. Acar A, Aksu H, Uluagac AS, Conti M (2018) A survey on homomorphic encryption schemes: theory and implementation. ACM Comput Surveys (Csur) 51(4):1–35
3. The Merriam-Webster's Dictionary Website. http://www.merriam-webster.com
4. Liddell HG, Scott R (1896) An intermediate Greek-English Lexicon: founded upon the seventh edition of Liddell and Scott's Greek-English Lexicon. Harper & Brothers, New York
5. Gentry C (2009) A fully homomorphic encryption scheme. Stanford University, California
6. Naseem S, Ahmed K (2017) Information protection in cognitive science. Int J Comput Sci Netw Secur (IJCSNS) 17(3):1

7. Zhang J-W, Xu G, Chen X-B, Chang Y, Dong Z-C (2023) Improved multiparty quantum private comparison based on quantum homomorphic encryption. Physica A 610:128397
8. Liang M (2013) Symmetric quantum fully homomorphic encryption with perfect security. Quantum Inf Process 12(12):3675–3687
9. Alagic G, Dulek Y, Schaffner C, Speelman F (2017) Quantum fully homomorphic encryption with verification. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3–7, 2017, Proceedings, Part I 23, pp. 438–467. Springer
10. Brakerski Z, Gentry C, Vaikuntanathan V (2014) (Leveled) fully homomorphic encryption without bootstrapping. ACM Trans Comput Theory (TOCT) 6(3):1–36
11. Sahai A, Waters B (2014) How to use indistinguishability obfuscation: deniable encryption, and more. In: Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing, pp. 475–484
12. Hoofnagle CJ, Van Der Sloot B, Borgesius FZ (2019) The European union general data protection regulation: what it is and what it means. Inf Commun Technol Law 28(1):65–98
13. Regev O (2009) On lattices, learning with errors, random linear codes, and cryptography. J ACM 56(6):1–40
14. Tambe-Jagtap SN (2023) A survey of cryptographic algorithms in cybersecurity: from classical methods to quantum-resistant solutions. SHIFRA 2023:43–52
15. Shor PW (1994) Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134. IEEE
16. Bernstein DJ, Lange T (2017) Post-quantum cryptography. Nature 549(7671):188–194
17. Childs AM (2001) Secure assisted quantum computation. arXiv preprint arXiv: quant-ph/0111046
18. Mosca M (2018) Cybersecurity in an era with quantum computers: will we be ready? IEEE Secur Privacy 16(5):38–41
19. Cheon JH, Kim A, Kim M, Song Y (2017) Homomorphic encryption for arithmetic of approximate numbers. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3–7, 2017, Proceedings, Part I 23, pp. 409–437. Springer
20. Halevi S, Shoup V (2014) Algorithms in Helib. In: Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17–21, 2014, Proceedings, Part I 34, pp. 554–571. Springer
21. Microsoft SEAL (release 4.1) (2023). https://github.com/Microsoft/SEAL. Microsoft Research, Redmond, WA
22. Jain N, Cherukuri AKA, Kamalov F (2024) Revisiting fully homomorphic encryption schemes for privacy-preserving computing. Emerging technologies and security in cloud computing. IGI Global, New York, pp 276–294
23. PALISADE Development Team (2021) PALISADE Lattice Cryptography Library. Retrieved from https://palisade-crypto.org/
24. Chillotti I, Gama N, Georgieva M, Izabachène M (August 2016) TFHE: Fast Fully Homomorphic Encryption Library. https://tfhe.github.io/tfhe/
25. Lattigo v5 (2023) Online: https://github.com/tuneinsight/lattigo. EPFL-LDS, Tune Insight SA
26. Doan TVT, Messai M-L, Gavin G, Darmont J (2023) A survey on implementations of homomorphic encryption schemes. J Supercomput 79(13):15098–15139
27. Gentry C, Sahai A, Waters B (2013) Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2013. Proceedings, Part I, pp. 75–92. Springer
28. Nielsen MA, Chuang IL (2010) Quantum computation and quantum information. Cambridge University Press, Cambridge
29. Preskill J (2018) Quantum computing in the NISQ era and beyond. Quantum 2:79
30. Horodecki R, Horodecki P, Horodecki M, Horodecki K (2009) Quantum entanglement. Rev Mod Phys 81(2):865
31. Barenco A, Bennett CH, Cleve R, DiVincenzo DP, Margolus N, Shor P, Sleator T, Smolin JA, Weinfurter H (1995) Elementary gates for quantum computation. Phys Rev A 52(5):3457
32. Shor PW (1999) Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev 41(2):303–332
33. Grover LK (1996) A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, pp. 212–219
34. Shor PW (1995) Scheme for reducing decoherence in quantum computer memory. Phys Rev A 52(4):2493
35. Broadbent A, Jeffery S (2015) Quantum homomorphic encryption for circuits of low t-gate complexity. In: Annual Cryptology Conference, pp. 609–629. Springer
36. Liang M, Yang L (2015) Quantum fully homomorphic encryption scheme based on quantum fault-tolerant construction. arXiv preprint arXiv:1503.04061
37. Brakerski Z (2018) Quantum FHE (almost) as secure as classical. In: Annual International Cryptology Conference, pp. 67–95. Springer
38. Mahadev U (2020) Classical homomorphic encryption for quantum circuits. SIAM J Comput 52(6):18–189
39. Gentry C, Sahai A, Waters B (2013) Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2013. Proceedings, Part I, pp. 75–92. Springer
40. Brakerski Z, Vaikuntanathan V (2014) Lattice-based FHE as secure as PKE. In: Proceedings of the 5th Conference on Innovations in Theoretical Computer Science, pp. 1–12
41. Alperin-Sheriff J, Peikert C (2014) Faster bootstrapping with polynomial error. In: Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17–21, 2014, Proceedings, Part I 34, pp. 297–314. Springer
42. Brakerski Z, Vaikuntanathan V (2014) Efficient fully homomorphic encryption from (standard) LWE. SIAM J Comput 43(2):831–871
43. Aaronson S, Christiano P (2012) Quantum money from hidden subspaces. In: Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing, pp. 41–60
44. Tan S-H, Kettlewell JA, Ouyang Y, Chen L, Fitzsimons JF (2016) A quantum approach to homomorphic encryption. Sci Reports 6(1):33467
45. Zeuner J, Pitsios I, Tan S-H, Sharma AN, Fitzsimons JF, Osellame R, Walther P (2021) Experimental quantum homomorphic encryption. NPJ Quantum Inf 7(1):25
46. Liu J, Li Q, Quan J, Wang C, Shi J, Situ H (2022) Efficient quantum homomorphic encryption scheme with flexible evaluators and its simulation. Designs Codes Cryptogr 90(3):577–591
47. Burhanuddin M (2023) Assessing the vulnerability of quantum cryptography systems to emerging cyber threats. SHIFRA 2023, 26–33. https://doi.org/10.70470/SHIFRA/2023/004
48. Liang M (2015) Quantum fully homomorphic encryption scheme based on universal quantum circuit. Quantum Inf Process 14(8):2749–2759
49. Zhang Y, Shang T, Liu J (2021) A multi-valued quantum fully homomorphic encryption scheme. Quantum Inf Process 20:1–25
50. Tham WK, Ferretti H, Bonsma-Fisher K, Brodutch A, Sanders BC, Steinberg AM, Jeffery S (2020) Experimental demonstration of quantum fully homomorphic encryption with application in a two-party secure protocol. Phys Rev X 10(1):1–25

51. Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement learning: a survey. J Artif Intell Res 4:237–285
52. Sutton RS, Barto AG et al (1999) Reinforcement learning. J Cognit Neurosci 11(1):126–134