# ZionFHE LitePaper

contact@zionfhe.ai

May 2025

# Contents

# 1. Abstract

In today's era of rapid AI development, data privacy and security issues are becoming increasingly prominent, emerging as major obstacles to AI commercialization. Traditional encryption methods can only secure data during storage and transmission; once computation is required, data must be decrypted into plaintext, thereby exposing it to privacy risks. Fully Homomorphic Encryption (FHE) enables computation directly on encrypted data, ensuring end-to-end data protection. Even when operations are performed on third-party platforms or in the cloud, the plaintext remains undisclosed, providing a comprehensive solution to AI data privacy challenges.

**ZionFHE** is the world's first solution that brings FHE into practical application. With its novel algorithmic architecture, ZionFHE has successfully overcome the performance bottlenecks of FHE, enabling AI models to be trained and executed entirely on encrypted data, thus ensuring the security of user information in the cloud and across third-party platforms.
ZionFHE's core products and solutions include:

- **Encrypted AI Model Inference:** Data remains encrypted throughout the process, and inference outputs are also encrypted, accessible only to the data owner.
- **GPU Acceleration and Performance Optimization:** Leverages GPU parallelism to significantly improve encrypted computation efficiency, narrowing the gap with plaintext operations.
- **AI Agents Ecosystem:** Builds FHE-based intelligent agent systems that support secure data sharing and inference across multiple endpoints (mobile, PC, cloud, etc.).
- **Cross-Platform Integration:** Provides standard APIs and developer tools for seamless integration with mainstream AI services, cloud platforms, and blockchain systems.

By the end of 2024, the ZionFHE team had successfully achieved encrypted inference of the Meta-Llama3.1-8B model, reaching a speed of **0.05 tokens per second** on a 48-core CPU—only two orders of magnitude slower than plaintext inference. With high-end GPU acceleration, this performance can reach **1 token per second**, which essentially meets commercial deployment requirements.

Beyond AI data privacy protection, ZionFHE also delivers low-energy, high-security cryptographic computation solutions for blockchain and distributed network scenarios. When combined with smart contracts, ZionFHE enables secure interactions with on-chain data and contract states, effectively addressing the long-standing privacy challenges of traditional blockchains.

The launch of ZionFHE marks the transition of FHE from theory to real-world commercial application, opening new opportunities for the convergence of AI and blockchain ecosystems. In the short term, the company is focusing on real-world applications of AI inference, secure data sharing, and privacy protection. Over the medium to long term, ZionFHE aims to build a

complete **FHE-AI ecosystem**, driving forward the evolution of privacy-preserving computation and distributed intelligence.

Our mission is to safeguard user privacy, accelerate AI adoption, and build a secure and trustworthy era of privacy-first AI.

# 2. Background

As AI rapidly evolves, data has become a vital resource driving global transformation, but it also brings heightened risks around security and privacy. Sensitive information used in AI training, inference, and sharing is highly vulnerable to misuse, as shown by cases like Samsung's internal data leakage, OpenAI's data management issues, and Apple's compliance challenges in Europe. These highlight the weak links in current AI privacy protection.

Traditional methods such as TLS and AES can secure data in transit and at rest, but during AI inference, data must be decrypted into plaintext, creating exposure risks. While techniques like federated learning and differential privacy help reduce leakage probability, they cannot fundamentally prevent access to information during computation.

Fully Homomorphic Encryption (FHE) provides a breakthrough solution: enabling computations directly on encrypted data, thus ensuring end-to-end privacy across transmission, storage, and inference—even on untrusted platforms.

Historically, FHE's high computational cost made it impractical, often thousands of times slower than plaintext operations. ZionFHE addresses this with an innovative algorithmic architecture that achieves encrypted AI inference at near-practical speeds for the first time. This breakthrough not only secures AI inference but also allows enterprises and individuals to leverage AI efficiently without compromising privacy, laying a strong foundation for real-world adoption.

# 3. Overview of Fully Homomorphic Encryption (FHE)

To achieve efficient computation without sacrificing data privacy, **Fully Homomorphic Encryption (FHE)** was developed. Often described as the *"Holy Grail"* of cryptography, FHE allows complex mathematical operations to be performed directly on encrypted data, enabling secure computation while data remains in an "invisible" state. This technology breaks through the limitations of traditional encryption, which can only protect data during storage and transmission, and offers a new solution for AI inference, privacy protection, and data sharing.

## 3.1 Concept and Basic Principles

The concept of homomorphic encryption (HE) was first proposed by Rivest et al. in 1978. Its core idea is to allow computations to be performed on encrypted data without decryption, such that the decrypted result is identical to the result obtained by performing the same operation on plaintext. In other words, given plaintext *m* and its ciphertext *E(m)*, after applying an operation *f* on the ciphertext, the decrypted result satisfies:

$$\mathbf{D(f(E(m))) = f(m)}$$

Early HE schemes only supported a single operation, either addition or multiplication, which is known as **Partial Homomorphic Encryption (PHE)**. In 1982, Goldwasser and Micali introduced the first scheme that supported limited additions. Over the following decades, researchers attempted to design schemes supporting both addition and multiplication, but the complexity of ciphertext structures and computational overhead prevented a breakthrough toward **Fully Homomorphic Encryption (FHE)**.

In 2009, Craig Gentry, in his doctoral dissertation, proposed the first complete FHE framework. He proved that through a method called **bootstrapping**, unlimited additions and multiplications could be performed on ciphertext. Gentry's scheme introduced the concept of **ciphertext refresh**, which reduces noise accumulation after each operation to maintain ciphertext stability and prevent errors from escalating.

Although Gentry's scheme demonstrated theoretical feasibility, its performance was extremely poor—each FHE operation could take minutes or longer, making it impractical. Over the past decade, numerous improvements have been proposed, including more efficient encoding methods, faster bootstrapping, and lower-complexity arithmetic. However, even the most advanced modern FHE implementations remain significantly slower than plaintext computation, hindering real-world adoption.

## 3.2 Improvements and Technical Evolution

Gentry's 2009 bootstrapping method established FHE's feasibility but was computationally prohibitive. Subsequent academic and industrial advancements focused on improving performance, with key milestones including:

- **LWE-based schemes:** In 2011, Brakerski and Vaikuntanathan (BV) introduced a scheme using noise management techniques to shorten bootstrapping time, though overall performance remained limited.
- **RLWE-based schemes:** In 2013, the BGV scheme mapped encryption operations into

ring structures, introducing modulus switching and vectorized computations. This reduced complexity and memory usage, though noise accumulation remained an issue.

- **TFHE scheme:** Introduced in 2016, TFHE optimized Boolean gate operations with fast bootstrapping and efficient key switching. It significantly improved Boolean computation efficiency but was limited to gate-level operations, making it unsuitable for large-scale floating-point and matrix computations.
- **CKKS scheme:** Proposed in 2017, CKKS enabled approximate floating-point arithmetic under encryption, making it practical for neural network inference and statistical analysis. Today, most AI-focused FHE implementations build upon CKKS. Companies such as Google and IBM continue to optimize CKKS, but challenges remain in cross-platform compatibility and scalability.

## 3.3 Modern Implementations and Application Bottlenecks

Despite significant progress in recent years, practical deployment of FHE still faces key bottlenecks:

- **Computational Complexity and Speed:** FHE operations (e.g., addition, multiplication, bootstrapping) involve intensive modular and polynomial arithmetic, typically 100–1000 times slower than plaintext operations—even on high-performance GPUs.
- **Memory and Storage Requirements:** FHE ciphertexts are typically 100–1000 times larger than plaintexts, creating substantial demands on memory and storage, particularly during bootstrapping where large intermediate results and keys must be stored.
- **Compatibility and Scalability:** Integrating FHE with existing AI frameworks (e.g., PyTorch, TensorFlow) requires extensive low-level adaptation due to its mathematical complexity and specialized hardware instructions, limiting large-scale deployment.
- **Bootstrapping Bottlenecks:** Even with TFHE and CKKS optimizations, bootstrapping operations can still take tens to hundreds of milliseconds, causing performance degradation in recursive or multiplication-heavy computations.

Against this backdrop, **ZionFHE** introduces a novel vectorized encoding strategy combined with GPU parallelization, significantly improving FHE computation speed and bringing performance closer to plaintext levels. In practical AI inference, ZionFHE achieves performance improvements of **one to two orders of magnitude**, enabling the realistic application of FHE across AI, privacy-preserving computation, and blockchain scenarios. This breakthrough opens a new path for FHE adoption in commercial and large-scale environments.

# 4. Core Technology of ZionFHE

ZionFHE adopts a novel high-performance Fully Homomorphic Encryption algorithm, **ZFHE**, whose security in the worst case can be reduced to solving the **Multivariate Quadratic (MQ) Problem**. By introducing **Lookup Tables (LUTs)** into ciphertext computations, ZFHE greatly improves the efficiency of homomorphic operations.

## 4.1 Overview of ZFHE Principles

Traditional cryptographic systems are defined by a triplet $\{E, D, KGen\}$, where $P$ denotes plaintext, $C$ denotes ciphertext, and $K$ denotes a key. The encryption operation is $C = E(P, K)$, the decryption operation is $P = D(C, K)$, and the key generation is $KGen(rand) = K$.

A fully homomorphic encryption system, by contrast, is defined by a quadruplet $\{E, D, KGen, Eval\}$, which introduces an additional operation, $Eval$, to enable computation on ciphertexts. Specifically, it satisfies:

1. For a set of plaintexts $P_i$, encryption yields ciphertexts $C_i = E(P_i, K), i \in [1, n]$;
2. An arbitrary function $F$ can be applied homomorphically to the ciphertexts, producing $C_{res} = Eval(F, \{C_1, C_2, \dots, C_n\})$;
3. Decryption of the result yields $P_{res} = D(C_{res}, K)$.
4. This result is equivalent to applying $P_i$ directly on plaintexts, i.e., $P_{res} = F(P_1, P_2, \dots, P_n)$.

On top of this structure, ZFHE introduces a **Lookup Table Generation** operation, $LUTGen$, which supports ciphertext computations by precomputing mappings for encrypted function evaluations. Similar to the "evaluation key" concept in [Lai2016], this allows secure and efficient ciphertext operations. Thus, ZFHE is defined by a quintuple$\{E, D, KGen, Eval, Dict\}$.

Here, $LUTGen(K) = G$ takes the secret key as input and outputs a lookup table $G$ Any ciphertext computation then requires participation of $G$, i.e., $C_{res} = Eval(F, \{C_1, C_2, \dots, C_n\}, G)$

**Ciphertext Construction**

ZFHE ciphertexts are constructed from **multivariate higher-order polynomials** with unknown functions. When the highest degree is 2, the underlying hardness is equivalent to the **MQ Problem**, formally expressed as:

$$P = \sum_{i=1}^{n} a_i \cdot f(x_i) \cdot y_i$$

The secret key $K$ consists of both the unknown function $f$ and a set of variables $C = \{A, X\}$,

$A = \{a_i | i \in I\}$, $X = \{x_i | i \in I\}$, $I = [1, n]$. A key innovation of ZFHE is incorporating the **expression of function** $f$ itself into the secret key. In implementation, several function templates can be defined, with both template codes and parameters serving as key material. Since ciphertexts are polynomial expressions combining vectors, multiplication of two ciphertexts raises the polynomial degree. Thus, ciphertext extension and reduction mechanisms are required.

### Lookup Table Design

The ZFHE lookup table is defined as:

$$G = \{g_1, g_2, h_1, h_2\}$$

$$\begin{cases} g_1(x_1, x_2, c) = \dfrac{f(x_1) + c \cdot f(x_2)}{f[h_1(x_1, x_2)]} \\ g_2(x_1, x_2) = \dfrac{f(x_1) \cdot f(x_2)}{f[h_2(x_1, x_2)]} \\ h_1(x_1, x_2), h_2(x_1, x_2) : R^2 \to R \\ h_1(x_1, x_2) \neq h_2(x_1, x_2) \neq x_1 \neq x_2 \end{cases}$$

Here, $g_1, g_2$ are multi-dimensional key–value mappings, implemented discretely with lookup tables rather than analytic expressions. Input variables serve as keys, while computed function values are stored as values. To ensure coverage of the domain, $f(x)$ can be defined with a limited domain or periodicity.

The mapping functions $h_1(x_1, x_2), h_2(x_1, x_2)$ are surjective from 2D real space to the 1D real subspace, ensuring domain traversal.

In essence, the lookup table functions much like a logarithm table: by precomputing discrete values of continuous functions, interpolation can provide fast approximations during runtime.

### Encryption

ZFHE encryption proceeds as follows:

1. Randomly sample two real numbers $x_1, x_2 \in Def$;

2. Randomly select $\alpha \in (0，1)$, and compute $P_1 = P \cdot \alpha$;
3. Compute coefficients $a_1 = \frac{P_1}{f(x_1) \cdot y_1}$, $a_2 = \frac{P - P_1}{f(x_2) \cdot y_2}$.

The resulting ciphertext is $C = \{a_1, x_1, a_2, x_2\}$.

Since three random variables are introduced, ciphertexts for identical plaintexts differ significantly across encryptions, providing strong resistance against statistical attacks.

### Decryption

Decryption is straightforward:

$$P = a_1 \cdot f(x_1) \cdot y_1 + a_2 \cdot f(x_2) \cdot y_2$$

### Ciphertext Addition

For two ciphertexts *C1* and *C2*, the addition results in another quadratic ciphertext. Using lookup tables, the result can be derived efficiently, ensuring closure under homomorphic addition.

$$\begin{aligned}
& C_1 + C_2 \\
& = [a_{11} \cdot f(x_{11}) \cdot y_1 + a_{12} \cdot f(x_{12}) \cdot y_2] + [a_{21} \cdot f(x_{21}) \cdot y_1 + a_{22} \cdot f(x_{22}) \cdot y_2] \\
& = a_{11} \cdot [f(x_{11}) + \frac{a_{21}}{a_{11}} \cdot f(x_{21})] \cdot y_1 + a_{12} \cdot [f(x_{12}) + \frac{a_{22}}{a_{12}} \cdot f(x_{22})] \cdot y_2 \\
& = a_{11} \cdot g_1(x_{11}, x_{21}, \frac{a_{21}}{a_{11}}) \cdot f[h_1(x_{11}, x_{21})] \cdot y_1 + a_{12} \cdot g_1(x_{12}, x_{22}, \frac{a_{22}}{a_{12}}) \cdot f[h_1(x_{12}, x_{22})] \cdot y_2 \\
& = a_{31} \cdot f(x_{31}) \cdot y_1 + a_{32} \cdot f(x_{32}) \cdot y_2 \\
& = C_3
\end{aligned}$$

### Ciphertext Multiplication

The multiplication process is more complex. The product of two quadratic ciphertexts yields a cubic ciphertext, which must then be reduced back to quadratic form through a degree-reduction operation. Since multiplication involves both addition and multiplication internally, it requires more lookup table queries than addition.

$$\begin{aligned}
& C_1 \cdot C_2 \\
& = [a_{11} \cdot f(x_{11}) \cdot y_1 + a_{12} \cdot f(x_{12}) \cdot y_2] \cdot [a_{21} \cdot f(x_{21}) \cdot y_1 + a_{22} \cdot f(x_{22}) \cdot y_2] \\
& = a_{11} \cdot a_{21} \cdot f(x_{11}) \cdot f(x_{21}) \cdot y_1^2 + [a_{11} \cdot a_{22} \cdot f(x_{11}) \cdot f(x_{22}) + a_{12} \cdot a_{21} \cdot f(x_{12}) \cdot f(x_{21})] \cdot y_1 \\
& \qquad \cdot y_2 \\
& \quad + a_{12} \cdot a_{22} \cdot f(x_{12}) \cdot f(x_{22}) \cdot y_2^2 \\
& = a_{11} \cdot a_{21} \cdot g_2(x_{11}, x_{21}) \cdot f[h_2(x_{11}, x_{21})] \cdot y_1^2 + [a_{11} \cdot a_{22} \cdot g_2(x_{11}, x_{22}) \cdot f[h_2(x_{11}, x_{22})] \\
& \quad + a_{12} \cdot a_{21} \cdot g_2(x_{12}, x_{21}) \cdot f[h_2(x_{12}, x_{21})]] \cdot y_1 \cdot y_2 + a_{12} \cdot a_{22} \cdot g_2(x_{12}, x_{22}) \cdot f[h_2(x_{12}, x_{22})] \\
& \qquad \cdot y_2^2 \\
& = a_{31} \cdot f(x_{31}) \cdot y_1^2 + [a'_{32} \cdot f(x'_{32}) + a''_{32} \cdot f(x''_{32})] \cdot y_1 \cdot y_2 + a_{33} \cdot f(x_{33}) \cdot y_2^2 \\
& = a_{31} \cdot f(x_{31}) \cdot y_1^2 + a'_{32} \cdot g_1(x'_{32}, x''_{32}, \frac{a'_{32}}{a''_{32}}) \cdot f[h_1(x'_{32}, x''_{32})] \cdot y_1 \cdot y_2 + a_{33} \cdot f(x_{33}) \cdot y_2^2 \\
& = a_{31} \cdot f(x_{31}) \cdot y_1^2 + a_{32} \cdot f(x_{32}) \cdot y_1 \cdot y_2 + a_{33} \cdot f(x_{33}) \cdot y_2^2 \\
& = C_4
\end{aligned}$$

# 4.2 Comparison with Traditional FHE Schemes

In the field of Fully Homomorphic Encryption (FHE), traditional schemes often face limitations such as low performance, severe noise accumulation, high bootstrapping costs, and insufficient support for complex models, making large-scale deployment difficult. ZionFHE, through an innovative architecture and computation optimizations, demonstrates significant advantages in encrypted inference efficiency, flexibility, stability, and compatibility.

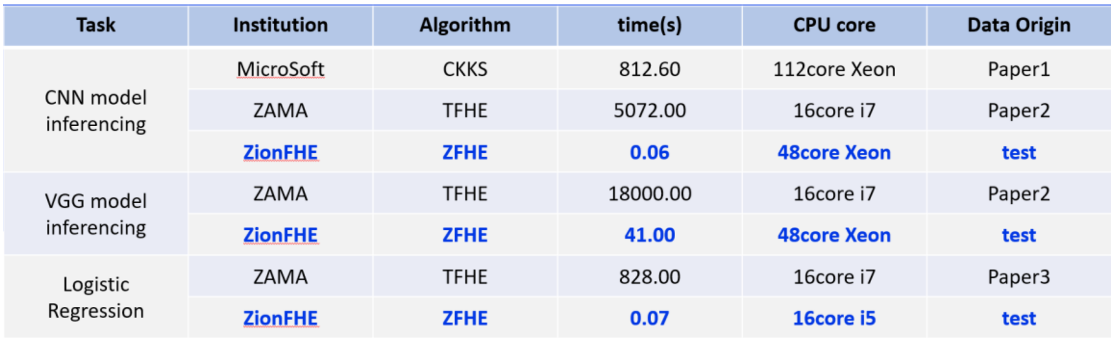
**Performance Advantages:** Traditional FHE schemes (e.g., CKKS and TFHE) typically require anywhere from tens of seconds to several hours for complex model inference. In contrast, ZionFHE, built on the ZFHE framework, leverages Ring-LWE homomorphism and dynamic modulus switching to effectively reduce computational redundancy and memory usage.

**Noise Management and Bootstrapping Optimization:** Noise accumulation during FHE operations often leads to decryption failures, with traditional schemes requiring frequent and costly bootstrapping. ZionFHE addresses this challenge by combining dynamic modulus switching with low-cost bootstrapping techniques (integrating FFT and sparse matrix optimizations), thereby significantly reducing noise growth. Additionally, its built-in intelligent monitoring system dynamically adjusts bootstrapping frequency, ensuring noise levels remain controllable during deep neural network inference.

**Adaptability and Compatibility:** ZionFHE adopts a modular design, offering seamless compatibility with mainstream AI frameworks such as TensorFlow and PyTorch. It provides specialized optimizations for different model types, including CNNs, RNNs, and Transformers. Furthermore, its cross-platform support (covering Intel, AMD, and Nvidia hardware) ensures performance differences remain within 10%, making it adaptable across diverse application scenarios.

**Computation Precision and Security:** By combining dynamic modulus switching with low-cost bootstrapping, ZionFHE keeps precision loss under 0.1%, ensuring outputs from complex models remain close to plaintext inference results. Meanwhile, its multi-layer Ring-LWE encryption combined with randomized perturbation strategies strengthens resistance against side-channel attacks and quantum computing, meeting international security standards.
Overall, ZionFHE outperforms traditional FHE schemes in performance, noise management, compatibility, precision, and security, delivering a reliable and efficient privacy-preserving solution for real-world AI inference.

## 4.3 Performance Benchmarks and Results



| Task | Institution | Algorithm | time(s) | CPU core | Data Origin |
|---|---|---|---|---|---|
| CNN model inferencing | MicroSoft | CKKS | 812.60 | 112core Xeon | Paper1 |
| | ZAMA | TFHE | 5072.00 | 16core i7 | Paper2 |
| | **ZionFHE** | **ZFHE** | **0.06** | **48core Xeon** | **test** |
| VGG model inferencing | ZAMA | TFHE | 18000.00 | 16core i7 | Paper2 |
| | **ZionFHE** | **ZFHE** | **41.00** | **48core Xeon** | **test** |
| Logistic Regression | ZAMA | TFHE | 828.00 | 16core i7 | Paper3 |
| | **ZionFHE** | **ZFHE** | **0.07** | **16core i5** | **test** |

**Figure 1: Comparison of ZFHE and Other FHE Schemes**

To validate ZionFHE’s practical performance, the team conducted comprehensive tests across a wide range of AI inference tasks, benchmarking against mainstream homomorphic encryption schemes including CKKS and TFHE. The results highlight ZionFHE’s significant advantages in inference speed, hardware compatibility, and scalability.

In **CNN (Convolutional Neural Network) inference tasks**, ZionFHE achieved outstanding performance, accelerating by 5–6 orders of magnitude compared to the other FHE algorithms. In more complex **VGG (Visual Geometry Group) network inference tasks**, ZionFHE continued to deliver excellent results. In **logistic regression inference tasks**, its advantage was demonstrated through higher throughput and lower latency.

# 5 ZionFHE Products and Solutions

As a next-generation provider of high-performance fully homomorphic encryption (FHE) technologies, ZionFHE’s product architecture is carefully designed to balance performance, flexibility, and security. The result is a complete ecosystem that spans from low-level encryption modules to high-level AI inference services.

## 5.1 Encrypted AI Model Inference

Performing AI inference under fully homomorphic encryption is one of ZionFHE’s core technical advantages. Traditional AI inference is typically executed in plaintext, which, while efficient, poses significant limitations in terms of data privacy and compliance. ZionFHE enables complex AI inference to be executed directly on encrypted data, without exposing original datasets or model parameters.

The encrypted inference workflow in ZionFHE includes four main stages: **Encrypted Input → Model Transformation → Homomorphic Inference → Decrypted Output.**

1. **Encrypted Input:** Before sending data into ZionFHE, users first encrypt it with ZionFHE's encryption module, transforming plaintext into ciphertext vectors. During encryption, modulus layering and noise barriers are introduced to ensure strong data security.
2. **Model Transformation:** In conventional AI environments, neural networks are built in frameworks such as PyTorch and executed on GPUs or CPUs as tensors. ZionFHE's built-in model compiler and optimizer automatically convert models from mainstream frameworks into a homomorphic computation format.
3. **Homomorphic Inference:** Once transformed, ZionFHE executes inference directly on encrypted data through parallel homomorphic addition, multiplication, and modulus switching operations. It supports multi-batch parallel inference, running tasks concurrently across multi-core CPUs and GPUs to improve throughput.
4. **Decrypted Output:** After encrypted inference is complete, ZionFHE decrypts the ciphertext result using the private key, producing the plaintext output. Because noise levels are controlled within safe thresholds via dynamic modulus adjustment and low-cost bootstrapping, the decrypted outputs maintain precision close to plaintext inference.

## 5.2 GPU Acceleration and Performance Optimization

ZionFHE is dedicated to maximizing the performance of homomorphic encryption (FHE) computations while maintaining the highest levels of security and privacy protection. To address the inherently computation-intensive nature of fully homomorphic encryption, ZionFHE integrates GPU parallel computing with a series of algorithmic optimization strategies, creating an efficient, low-latency encrypted inference system. The core strategy lies in leveraging the massive parallelism of GPUs to accelerate modular arithmetic and polynomial operations—traditionally burdensome on CPUs—through vectorization, batch processing, and pipelining techniques. This approach enables performance on large-scale encrypted data processing tasks that approaches, as closely as possible, the speed of plaintext computation.

First, ZionFHE stores encrypted data in vector or tensor form directly in GPU memory, fully exploiting the GPU's parallel processing advantages. By using parallel computing frameworks such as CUDA, the system can simultaneously process tens of thousands of encrypted elements, enabling batch homomorphic operations. This vectorized approach significantly reduces the latency of individual encrypted computations while also minimizing communication overhead caused by frequent data transfers. Furthermore, the system employs multi-stream pipelining to schedule data transfer, kernel execution, and intermediate result processing in parallel, achieving seamless coordination across stages and further improving throughput.

At the algorithmic level, to handle dense matrix operations and convolutions common in deep learning inference, the system introduces sparse matrix and block computation techniques. By identifying sparsity and redundancy within data, it reduces both memory consumption and computational load, thereby enhancing overall speed. ZionFHE further leverages sparse matrix multiplication and pipelined scheduling strategies to dramatically lower the cost of lookup table queries and multiplications, mitigating the sharp performance degradation that typically occurs in deep neural networks due to repeated multiplications.

Overall, ZionFHE's GPU acceleration and performance optimization strategy is a multidimensional, multi-layered solution. Through tight hardware–software co-optimization, ZionFHE not only shortens encrypted computation time significantly but also achieves efficient inference for complex deep learning models such as CNNs and Transformers—without compromising the inherent security of homomorphic encryption. In practice, optimized encrypted inference speeds have reached one-tenth of plaintext computation or higher, providing commercially viable solutions for highly privacy-sensitive domains such as finance, healthcare, and government services. In summary, ZionFHE's GPU acceleration and algorithmic optimizations not only overcome the traditional performance bottlenecks of FHE but also establish a robust technical foundation for large-scale privacy-preserving computation in the future.

# 6 Company Background and Ecosystem

## 6.1 ZionFHE Company Background

ZionFHE is a high-tech company dedicated to the research and commercialization of Fully Homomorphic Encryption (FHE). The company's mission is to provide high-security, high-performance encryption solutions for global industries including artificial intelligence (AI), privacy-preserving computation, and blockchain. The founding vision of ZionFHE stems from the urgent need to safeguard data privacy: in today's increasingly digital and intelligent era, vast amounts of sensitive data are at constant risk of being stolen or misused during transmission, storage, and processing. Traditional encryption methods cannot provide complete privacy protection in complex scenarios such as AI inference, model training, or multi-party collaborative computation, which creates the need for breakthrough innovation.

## 6.2 Developer Ecosystem

ZionFHE recognizes that technological breakthroughs alone are not enough to meet the broad demands of privacy-preserving computation. The company therefore adopts an "open and collaborative" strategy to accelerate the development of the global FHE ecosystem.

- **Open APIs and SDKs**: Provides standardized APIs and cross-platform SDKs, enabling developers to easily integrate FHE capabilities into AI, blockchain, and cloud applications.

- **Technical Documentation and Example Projects**: Publishes comprehensive documentation, user guides, and sample projects to help developers quickly master homomorphic encryption.
- **Online Community and Forums**: Builds a global developer community supporting Q&A, knowledge sharing, and plugin exchanges, encouraging innovation and the creation of new applications.

# 7 Roadmap and Future Outlook

In an era of rapidly growing demand for privacy-preserving computation, ZionFHE is committed to advancing FHE from laboratory research into large-scale commercial applications. Based on an evaluation of current technical maturity, market feedback, and industry trends, ZionFHE has defined a roadmap covering short-term product deployment and pilot projects, mid-to-long-term technology iterations and feature expansion, and broader industry prospects.

In the near term, ZionFHE will focus on validating the feasibility and commercial value of FHE in real-world business scenarios. Key objectives include:

1. Achieving encrypted inference throughput of **over 1 token per second** on medium-scale models (e.g., Llama 3.1 8B) using a single GPU or small GPU clusters, and reaching **10–30 tokens/s** on an **8×H100 cluster**.
2. Transforming laboratory research into deliverable prototypes, packaged into APIs/SDKs with monitoring, logging, and fault recovery features, lowering the barrier for enterprise adoption.
3. Launching pilot projects in high-privacy-demand sectors such as finance, healthcare, and government, to validate compliance, security, and user experience—paving the way for broader rollout.

# 8 Conclusion

This white paper has presented ZionFHE's systematic roadmap and innovations in Fully Homomorphic Encryption (FHE), spanning technical foundations, product architecture, pilot deployments, and ecosystem development. As big data and AI technologies advance, the demand for robust data security and privacy protection across industries continues to rise. FHE, as a core technology of next-generation privacy-preserving computation, is poised for exponential growth.

Through deep algorithmic optimizations, GPU acceleration, and advanced techniques such as dynamic modulus switching and low-cost bootstrapping, ZionFHE achieves encrypted computation speeds approaching plaintext performance. This provides practical privacy-preserving solutions for complex scenarios in cloud computing, blockchain, and AI inference. Its dual-flow architecture design—supporting both plaintext and ciphertext pipelines—combined with modular components, offers enterprises and developers flexible and efficient deployment options.

Leveraging its core team's expertise in cryptography, AI, and blockchain, ZionFHE is building an ecosystem spanning industries with stringent privacy demands, such as finance, healthcare, government, and law. Through open APIs and SDKs, ZionFHE provides developers and partners with direct access to FHE technology, while short-term pilots and commercialization efforts lay the foundation for future large-scale adoption.

Looking ahead, as global data privacy regulations tighten and AI and blockchain technologies continue to evolve, FHE will find increasingly broad application. ZionFHE will remain at the forefront of this transformation, driving ongoing innovation and ecosystem growth, and helping to build a secure, compliant, and trustworthy digital future.