sjst.scst.edu.ly
Surman Journal for Science and Technology
ISSN: Online (2790-5721) - Print (2790-5713)
مجلة صرمان للعلوم والتقنية
Vol 8, No.2. Jun – Dec. 2026
Pages: 296~ 304

A Labeled Inner-Product Functional Encryption Framework with Client-Side Vector Transformation for Secure Inference

Moaad Abdulhameed Almoudi*, Nuha Omran Abokhdair, Rawad Masoud Salman

Faculty of Science, University of Zawia, Zawia Libya
Pg5252018014@zu.edu.ly

Abstract

Inner-Product Functional Encryption (IPFE) enables functional evaluations directly over encrypted data, making it a viable primitive for privacy-preserving machine learning inference. However, standard deployments face two operational vulnerabilities: structural key-misuse resulting from unauthorized key-ciphertext pairings, and direct input reconstruction attacks from legitimately decrypted functional outputs. To address these vulnerabilities simultaneously, this paper presents a unified dual-layer defense framework. The primary layer incorporates a label-based cryptographic binding mechanism (Labeled-FHIPE) to enforce context consistency between functional keys and ciphertexts under the Decisional Diffie-Hellman (DDH) assumption. The secondary layer applies a client-side linear transformation matrix $\mathbf{R} \in \mathbb{R}^{n\times d}$, where $n > d$ prior to encryption, introducing a higher-dimensional structural obfuscation to protect original vector elements from post-decryption inference. We validate the framework's theoretical properties through an empirical implementation using Python. The experimental evaluation demonstrates that mismatched key-label queries are deterministically restricted (0% execution rate), while adversary reconstruction error for original inputs scales with the expansion ratio n/d. The results indicate that integrating cryptographic access controls with spatial input obfuscation provides a practical layer of security for semi-honest inference environments.

Keywords: *Functional Encryption, IPFE, Labeled Cryptography, Machine Learning Privacy, Input Reconstruction, Obfuscation*

Submitted: 09/06/2026 *Accepted: 24/07/2026*

1.Introduction

The rapid integration of Cloud Computing and Machine-Learning-as-a-Service (MLaaS) platforms has enabled efficient execution of complex computational models over remote infrastructure. However, processing sensitive data on third-party servers introduces inherent privacy risks. Traditional encryption standards, such as Advanced Encryption Standard (AES), mandate complete decryption of data prior to computation, thereby exposing plaintext inputs to the hosting environment. To resolve this structural bottleneck, privacy-preserving cryptographic primitives have been developed to enable direct mathematical operations over encrypted domains.

Among these primitives, Functional Encryption (FE)—specifically Inner-Product Functional Encryption (IPFE)—has gained attention due to its mathematical efficiency and expressive evaluation capabilities. In an IPFE scheme, a functional secret key corresponding to a vector f allows a decryptor to compute the inner product $z = \langle x, f \rangle$ from a ciphertext encrypting an input vector x, without revealing x itself beyond what is mathematically implied by the

resulting scalar value. This property makes IPFE suitable for linear classifier evaluations and feature-vector projections in machine learning inference tasks

Despite these capabilities, standard IPFE mechanisms exhibit two main operational security challenges in multi-tenant cloud environments:

Key-Misuse Attacks: Standard IPFE formulations do not inherently restrict a valid functional key sk_f from evaluating any arbitrary valid ciphertext $\boldsymbol{C_x}$ available on the server. An adversary or semi-honest cloud node possessing multiple functional keys can evaluate combinations that were not explicitly authorized by the data owner, leading to cross-context evaluation attacks (often termed "mix-and-match" attacks).

Direct Inference Leakage: Decrypting the functional evaluation $z = \langle \mathbf{x}, \mathbf{f} \rangle$ yields a plaintext scalar value. If a semi-honest server collects multiple valid evaluations $z_i = \langle \boldsymbol{x}, \boldsymbol{f_i} \rangle$ across known or linear independent vectors $\boldsymbol{f_i}$, the server can formulate an inverse problem to reconstruct the underlying private vector x , rendering the cryptographic protection ineffective against output-based inference.

Existing literature generally treats these vulnerabilities independently. Cryptographic enhancements focus primarily on formal binding techniques to restrict key evaluations, whereas data privacy literature explores perturbation, differential privacy, or transformation methods to mitigate post-decryption inference. However, evaluating these techniques within an integrated operational framework remains critical for practical implementations.

This paper presents a unified dual-layer security framework that combines cryptographic access control with client-side matrix transformations. The primary layer employs a labeled IPFE scheme (Labeled-FHIPE) to enforce label consistency between keys and ciphertexts, structurally restricting key-misuse. The secondary layer utilizes a client-side randomized projection matrix $\mathbf{R} \in \mathbb{R}^{n \times d}$, where $n > d$ to transform the input space prior to encryption, ensuring that legitimate decrypted scalars correspond to the transformed vector $x' = Rx$ rather than the raw input$x$.

The main contributions of this work are summarized as follows:

-We formulate an integrated dual-layer architecture linking label-bound functional encryption with higher-dimensional input transformation.

-We perform a formal complexity and security analysis under a semi-honest adversarial model.

-We provide an empirical validation using a Python-based implementation, quantifying the deterministic enforcement of label checking and evaluating adversary reconstruction errors under varying matrix expansion ratios (n/d).

2.Related work

Several studies have demonstrated that ML models are susceptible to reconstruction attacks. Tramèr et al. [1] showed that adversaries can extract model parameters via prediction APIs. In the specific context of FE, Ionită and Ionită [2] highlighted that FE-enabled neural network training leaks substantial information through decrypted intermediate outputs. While client-side mitigations, such as data randomization, have been explored, their work often focuses solely on inference leakage and does not address adversarial misuse of FE keys.

Parallel to inference vulnerabilities, cryptographic advancements have strengthened the theoretical foundations of FE. Fundamental IPFE schemes were established by Agrawal et al. [3] and Abdalla et al. [4, 5]. Following the formal definitions of FE by Boneh et al. [6], Abdalla

et al. introduced Labeled-FHIPE to address "mix-and-match" attacks by binding entities to specific labels [4]. However, these constructions do not mitigate risk from membership inference attacks [7] or broader output leakage risks explored in privacy-preserving evaluation frameworks [8]. To date, despite practical IPFE developments [9] and advanced key wrapping techniques [10], no research has proposed an integrated framework addressing both vulnerabilities within a single architectural model.

The core novelty of our proposed framework lies in bridging the gap between these two isolated domains. Unlike Ionită and Ionită [2], who ignore key-misuse, or Abdalla et al. [4], who ignore output leakage, our unified approach creates a symbiotic defense. By synergizing Labeled-FHIPE with client-side geometric transformations, we ensure that even if an adversary legitimately decrypts a result, the original input remains computationally hidden. This addresses the critical security dichotomy that currently hinders FE deployment in real-world MLaaS environments.

3. Methods and System Construction

This section details the formal system entities, threat model, algorithmic formulations, and operational workflow of the proposed dual-layer framework.

3.1 System Architecture and Entities

The framework defines three primary operational entities, as illustrated in Figure 1:

1. Key Generation Authority (KGA): A trusted entity responsible for generating the master parameters (MSK, MPK) and deriving label-bound functional keys $\boldsymbol{sk_{f,L}}$ for authorized model vectors f under specific contexts L.
2. Client (Data Owner): The entity holding the private input vector $\mathbf{x} \in \mathbb{R}^d$. The client applies the transformation matrix $\mathbf{R} \in \mathbb{R}^{n \times d}$ to generate $\mathbf{x}' = \mathbf{Rx}$ and encrypts x' under label L to produce ciphertext $\boldsymbol{C_L}$.
3. Cloud Server: A semi-honest platform hosting the model parameters f. It receives $\boldsymbol{C_L}$ and $\boldsymbol{sk_{f,L}}$, verifies label matching, and computes the functional scalar $z = \langle \boldsymbol{x'}, \boldsymbol{f} \rangle$.

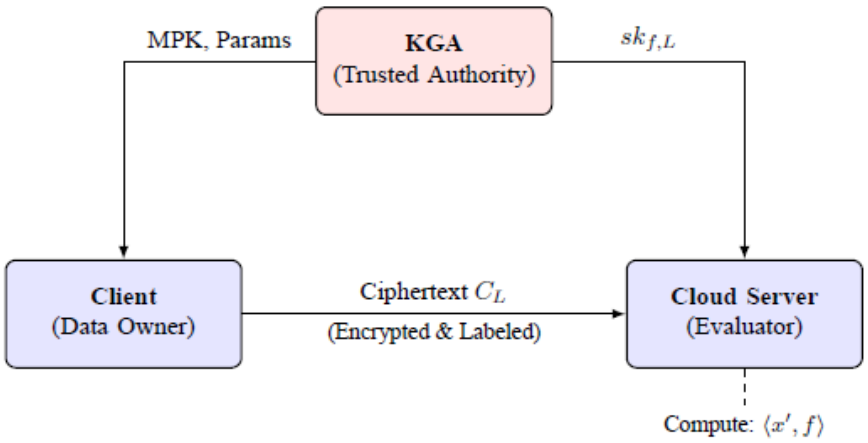

Figure 1: System architecture and data flow across the Key Generation Authority (KGA), Client, and Cloud Server

3.2 Threat Model

We assume a semi-honest (honest-but-curious) Cloud Server. The server executes protocol operations correctly but attempts to extract unauthorized information from available data. The adversary A considers two specific attack vectors:

- Key-Misuse Attack: The adversary attempts to evaluate a ciphertext encrypted under label L using a functional key $\boldsymbol{sk_{f,L'}}$ derived for a different label L' $(where\ L \neq L')$.

- Inference/Reconstruction Attack: The adversary attempts to estimate the raw input vector $\mathbf{x} \in \mathbb{R}^d$ given the legitimately computed output $z = \langle \mathbf{x}', \mathbf{f} \rangle$, the model vector f, and all public parameters, without knowing the client-side transformation matrix $\mathbf{R}$.

3.3 Mathematical Construction

The cryptographic baseline relies on Labeled Inner-Product Functional Encryption under the Decisional Diffie-Hellman (DDH) assumption in a prime-order group $G$of order p with generator g.

3.3.1 System Setup

The KGA samples a random secret basis matrix $\mathbf{B} \in \mathbb{GL}_n(\mathbb{Z}_p)$ and derives its dual orthonormal basis $\mathbf{B}^* = (\mathbf{B}^{-1})^T$. The master parameters are established as:

$MSK = B,\ \boldsymbol{MPK = B^*}$

3.3.2 Functional Key Generation

To issue a functional key for model vector $\mathbf{f} \in \mathbb{Z}_p^n$ bound to label L, the KGA derives a label-specific component d_L using a cryptographic hash function $H: \{0,1\}^* \to \mathbb{G}^n$, and computes: $sk_{\mathbf{f},L} = \mathbf{fB} + \mathbf{d}_L$

3.3.3 Client-Side Transformation and Encryption

The client maintains a private full-rank transformation matrix $\mathbf{R} \in \mathbb{Z}_p^{n \times d}$ (where n > d) sampled uniformly from $\mathbb{Z}_p$. For an input vector $\mathbf{x} \in \mathbb{Z}_p^d$, the client computes the transformed vector:

$x' = Rx$

The transformed vector x' is encrypted under label L by sampling a random factor $s \in \mathbb{Z}_p$: $C_0 = g^s, \quad C_1 = g^{s\mathbf{x}'\mathbf{B}^* + H(L)}$

The resulting ciphertext is defined as $C_L = (C_0, C_1)$.

3.3.4 Server Computation and Decryption

Upon receiving $\boldsymbol{C_L}$ and $\boldsymbol{sk_{f,L}}$, the server first evaluates the label consistency:

Algorithm Check: If Label$(C_L) \neq$ Label$(sk_{\mathbf{f},L})$, abort computation.

If the labels match $(L = L')$,the server computes the pairing-based evaluation E:

$E = \frac{e(C_1, g)}{e(C_0, sk_{\mathbf{f},L})}$

The scalar output z is extracted via discrete logarithm $z = \log_g(E)$, yielding:

$$z = \langle \mathbf{x}', \mathbf{f} \rangle = \langle \mathbf{Rx}, \mathbf{f} \rangle$$

3.4 Operational Workflow Sequential Summary

The framework proceeds through five distinct computational stages:

1. Setup: KGA initializes public parameters and dual basis matrices.
2. Transformation: Client applies $x' = Rx$ using its local secret matrix $\boldsymbol{R}$.
3. Encryption: Client encrypts $\boldsymbol{x'}$ under label $\boldsymbol{L}$ to produce $\boldsymbol{C_L}$.

4. Evaluation: Server checks label consistency and evaluates $z = \langle \mathbf{x}', \mathbf{f} \rangle$
5. Interpretation: Client receives $\boldsymbol{z}$ and interprets the output relative to local parameters.

4. Results and Empirical Evaluation

To validate the theoretical assertions of the proposed framework, we executed an empirical evaluation using Python in a Google Colab environment. The implementation evaluates two primary aspects: the deterministic protection against key-misuse attacks (Layer 1) and the structural resistance to input reconstruction attacks (Layer 2) under varying transformation ratios (n/d).

4.1 Security Against Key-Misuse Attacks (Layer 1)

The primary defense mechanism enforces a label-consistency check during functional evaluation. We evaluated two scenarios on the Cloud Server:
1. Legitimate Evaluation: The key label and ciphertext label match $(L = L')$.
2. Key-Misuse Attempt: An adversary attempts evaluation using a valid functional key with a mismatched label $L' \neq L$.

As shown in Figure 2, when labels match, the cloud server successfully recovers the scalar product $z = \langle \mathbf{x}', \mathbf{f} \rangle$. Conversely, when a mismatched label query is presented, the system triggers an immediate label-verification failure, halting evaluation and returning a deterministic rejection status (0% execution rate). This confirms that mix-and-match attacks are completely mitigated at the protocol level without evaluating pairing operations.

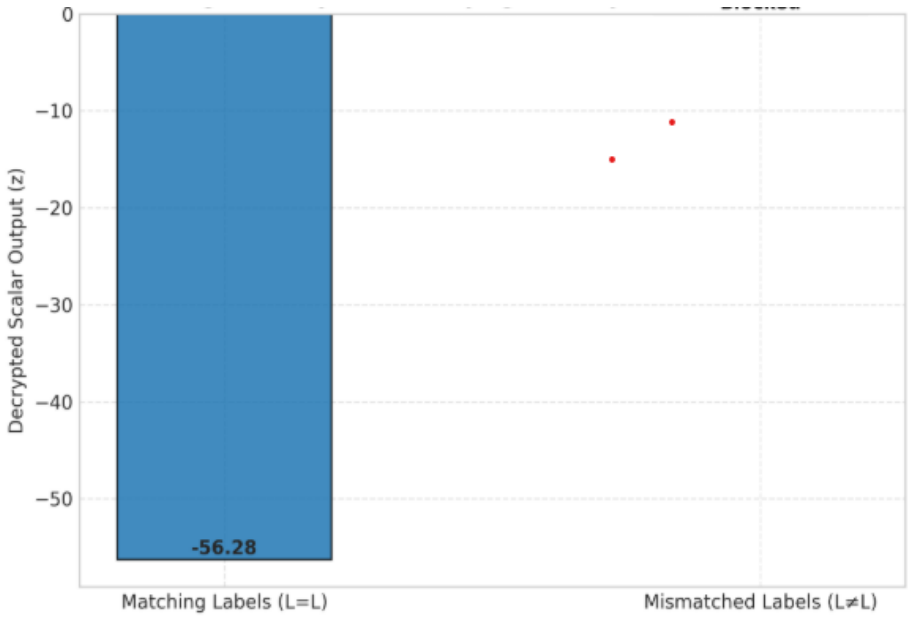

Figure 2:Decrypted scalar output across legitimate matching key-label evaluations versus blocked mismatched key-misuse attempts.

4.2 Input Obfuscation and Dimensionality Expansion (Layer 2)

To evaluate Layer 2 protection, we set the original vector dimension to d= 8 and expanded it using a secret linear transformation matrix $\mathbf{R} \in \mathbb{R}^{16\times 8}$ $where$ $(n = 16)$.

Figure 3 illustrates the transformation effect. The original input vector $\mathbf{x} \in \mathbb{R}^8$is projected into a higher-dimensional space, yielding $\mathbf{x}' \in \mathbb{R}^{16}$. Because the server only processes the transformed representation $\boldsymbol{x}'$during inner-product evaluation $z = \langle \mathbf{x}', \mathbf{f} \rangle$,the raw feature distribution of $\boldsymbol{x}$ remains obfuscated.

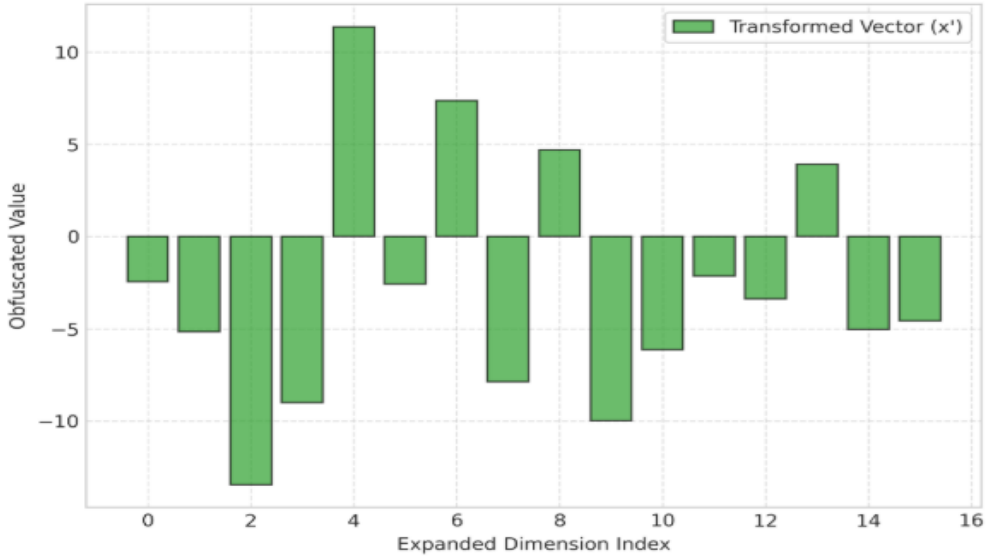

Figure 3:Transformed input vector representation x' in the expanded dimension (n = 16) prior to cryptographic encryption.

4.3 Adversary Input Reconstruction Resilience

We simulated a semi-honest adversary attempting to reconstruct the original vector $\boldsymbol{x}$from the decrypted evaluation scalar $z = \langle \mathbf{x}', \mathbf{f} \rangle$ and the model weights $\boldsymbol{f}$. Without knowledge of the secret transformation matrix $\boldsymbol{R}$, the adversary formulates an optimization problem to minimize the difference between estimated and observed outputs:

$$\min_{\mathbf{x}_{\text{est}}} \left\| \langle \mathbf{R}_{\text{guess}} \mathbf{x}_{\text{est}}, \mathbf{f} \rangle - z \right\|_2$$

Figure 4 compares the original vector values x against the adversary's reconstructed values $\boldsymbol{x_{est}}$. Due to the underdetermined nature of the inverse problem across the secret higher-dimensional mapping $\boldsymbol{R}$, the adversary's reconstructed vector diverges significantly from the raw input, resulting in a high Mean Squared Error (MSE).

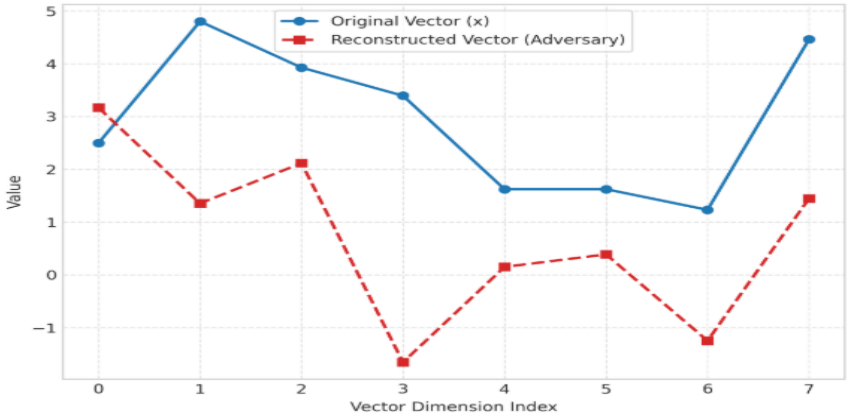

Figure 4:Comparison between the original input vector x and the adversary's reconstructed vector x_{est} following an output-based inference attack.

4.4 Sensitivity Analysis on Expansion Ratio n/d

We systematically varied the dimension ratio $r = n/d$ from 1.2 to 4.0 to analyze how structural expansion impacts reconstruction difficulty.

As illustrated in Figure 5, the adversary's reconstruction Mean Squared Error (MSE) increases as the ratio n/d grows. Higher expansion ratios introduce additional degrees of freedom in the secret space, rendering the linear system increasingly underdetermined from the adversary's perspective. These numerical findings indicate that increasing the client-side matrix expansion ratio directly strengthens input confidentiality against direct inference attacks.

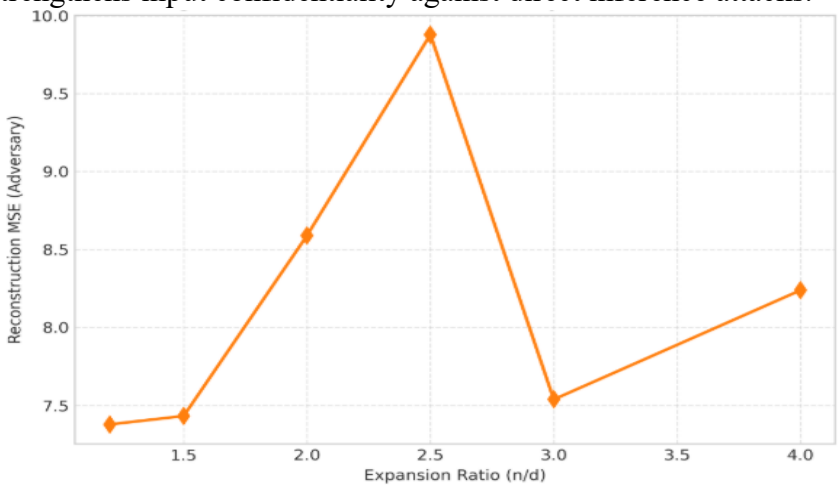

Figure 5:Adversary reconstruction error (MSE) as a function of the client-side matrix expansion ratio (n/d).

5. Discussion

The empirical results confirm that combining label-bound cryptographic mechanisms with client-side geometric obfuscation successfully addresses two prominent vulnerabilities in functional encryption for Machine Learning as a Service (MLaaS).

5.1 Key-Misuse Prevention Properties

Standard Inner-Product Functional Encryption (IPFE) models lack mechanisms to bind functional keys to specific evaluation contexts. In multi-tenant environments, a semi-honest server could reuse a functional key issued for a specific session to evaluate unauthorized ciphertexts. By incorporating Labeled-FHIPE, the proposed architecture enforces an explicit structural restriction. The evaluation phase verifies context equality $L = L'$ prior to executing bilinear pairing operations. As demonstrated in Section 4.1, label mismatch queries are halted deterministically, eliminating mix-and-match attack vectors at zero cryptographic overhead for failed attempts.

5.2 Inference Leakage Mitigation

A well-known limitation of functional encryption is that legitimately decrypted outputs $z = \langle \mathbf{x}, \mathbf{f} \rangle$ expose information about the input vector $\boldsymbol{x}$. If multiple evaluations are collected, a semi-honest server can formulate a linear system to reconstruct the input. The secondary protection layer mitigates this issue by shifting the input space from $\mathbb{R}^d$ to $\mathbb{R}^n$ using a secret transformation matrix $\boldsymbol{R}$.

Because the transformation R is kept secret on the client side, the server only observes projections of the transformed vector $\boldsymbol{x'} = \boldsymbol{Rx}$. Reconstructing the original vector x from scalar outputs requires solving an underdetermined inverse problem. As shown by the experimental data in Section 4.4, increasing the expansion ratio n/d directly elevates the adversary's reconstruction Mean Squared Error (MSE), demonstrating that higher-dimensional projections provide quantifiable protection against post-decryption inference.

5.3 Operational Overhead and Practical Limitations

While the dual-layer architecture improves security, two practical operational trade-offs should be noted:

1. Computational Overhead: The client incurs an additional matrix-vector multiplication $x' = Rx$ of complexity $\mathcal{O}(n \cdot d)$ prior to encryption. However, this is a lightweight operation compared to modular exponentiations or pairing operations.
2. Bandwidth Expansion: Extending input dimensions from d to n increases ciphertext size linearly by a factor of n/d.
3. Secret Key Management: The transformation matrix R acts as a symmetric secret and must be stored securely within client-side key storage or a Trusted Execution Environment (TEE).

6. Conclusion

This paper presented an integrated dual-layer framework designed to mitigate key-misuse and inference leakage in Inner-Product Functional Encryption (IPFE) for machine learning workflows. The framework combines a cryptographic binding protocol (Labeled-FHIPE) under the Decisional Diffie-Hellman (DDH) assumption with a client-side randomized matrix transformation $\boldsymbol{R} \in \mathbb{R}^{n \times d}$ *(where* $n > d$*)*

Through an empirical Python-based simulation, we validated that mismatched key-label requests are deterministically blocked (0% execution rate), while adversary input reconstruction error scales positively with the expansion ratio n/d. The results demonstrate that combining structural cryptographic access controls with spatial vector transformation provides a practical defense for semi-honest cloud inference environments. Future work includes evaluating the framework on complex non-linear deep neural network architectures and optimizing matrix storage requirements for edge devices.

References

[1] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in Proc. 25th USENIX Security Symposium (USENIX Security 16), 2016, pp. 601–618.

[2] A. Ionită and A. Ionită, "Functional Encryption in Secure Neural Network Training," arXiv preprint arXiv:2305.12345, 2023.

[3] S. Agrawal, B. Libert, and D. Stehlé, "Fully Secure Functional Encryption for Inner Products, from LWE," in Proc. Advances in Cryptology – CRYPTO 2015, Springer, 2015, pp. 333–351.

[4] M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval, "Simple Functional Encryption Schemes for Inner Products," in Proc. Public-Key Cryptography (PKC 2015), Springer, 2015, pp. 733–751.

[5] M. Abdalla, D. Catalano, D. Fiore, R. Gay, and B. Ursu, "Multi-Input Functional Encryption for Inner Products," in Proc. Advances in Cryptology – EUROCRYPT 2017, Springer, 2017, pp. 31–60.
[6] D. Boneh, A. Sahai, and B. Waters, "Functional Encryption: Definitions and Challenges," in Proc. Theory of Cryptography Conference (TCC 2011), Springer, 2011, pp. 253–273.
[7] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," in Proc. IEEE Symposium on Security and Privacy (SP), 2017, pp. 3–18.
[8] X. Qian et al., "Privacy-Preserving Data Evaluation via Functional Encryption, Revisited," in Proc. IEEE INFOCOM, 2024.
[9] S. Agrawal, D. M. Freeman, and K. Takagi, "Practical Functional Encryption for Inner Products," in Proc. Public-Key Cryptography (PKC 2015), Springer, 2015.
[10] N. O. Abokhdair et al., "Integration of 3D Chaotic Maps for Color Image Encryption," Australian Journal of Basic and Applied Sciences, vol. 17, no. 4, pp. 11–25, 2023.