Federated Gaussian Process Regression with Orthogonal Feature Encryption and Key-Based Access Control

Md. Rashedul Islam[1], Jannatul Ferdous Akhi[2], Takayuki Nakachi[3]
Graduate School of Engineering and Science, University of the Ryukyus, Okinawa, Japan[1,2]
Information Technology Center, University of the Ryukyus, Okinawa, Japan[3]

***Abstract*—Federated learning (FL) makes it possible to train models across distributed data sources without collecting raw data in one place. However, even in federated settings, trained models may still leak sensitive information at inference time. This problem is particularly evident for Gaussian Process regression (GPR), where predictive uncertainty is explicitly returned and can differ between training and non-training samples. Such differences can be exploited for membership inference. In this work, we examine inference-time privacy and robustness in federated GPR by focusing on the behavior of predictive variance. To enable scalable training, we employ a Random Fourier Feature approximation together with an Alternating Direction Method of Multipliers (ADMM) based distributed optimization scheme. On top of this learning framework, we apply key-dependent orthogonal feature transformations that enable multi-key inference time access control. When inference is performed using the correct key, prediction accuracy and uncertainty behavior remain close to those of plaintext federated GPR. When incorrect or mismatched keys are used, prediction errors increase sharply and predictive variance becomes uniformly large. Experimental results show that this variance inflation removes the usual gap between training and unseen samples, reducing the effectiveness of variance-based membership inference. Importantly, this effect arises without adding noise or relying on cryptographic operations. These findings suggest that predictive uncertainty can play a practical role in enforcing inference-time access control and improving privacy robustness in federated Gaussian Process models.**

***Keywords*—Gaussian process; differential privacy; Random Unitary Transformation; membership inference attack; machine learning; federated learning**

I. Introduction

Concerns regarding how large organizations collect, process, and utilize personal data have increased significantly in recent years. High-profile incidents involving data misuse have intensified public awareness and highlighted the need for accountability and responsible data governance. In response, regulatory frameworks such as the General Data Protection Regulation (GDPR), which came into effect in the European Union in 2018, have introduced strict legal requirements on the collection, storage, and sharing of personal data [1]. These regulations pose challenges for traditional machine learning pipelines, which typically assume centralized data access and unrestricted data sharing. As a result, there has been growing interest in decentralized learning paradigms that can operate under such constraints. Federated learning (FL) has emerged as a promising solution by enabling collaborative model training across distributed data sources without requiring raw data to be transferred to a central server [2]. Recent advances have further explored privacy-preserving mechanisms and system-level challenges in federated settings [3], [4], [5].

FL has been applied in various domains, including edge computing, intelligent transportation systems, and healthcare applications [6], [7], [8], [9], [10], [11], [12]. However, studies have shown that decentralizing data alone is not sufficient to guarantee privacy. Even in federated settings, trained models may still leak sensitive information through their outputs, giving rise to privacy risks such as membership inference attacks [13], [7].

Gaussian Process Regression (GPR) models present additional challenges in this setting. Unlike many deterministic learning methods, GPR explicitly reports predictive uncertainty as part of its output. In practice, the variance produced by a GPR model is often lower for inputs that are close to the training data than for inputs that are genuinely unseen. This behavior is a natural consequence of the probabilistic structure of Gaussian Processes, but it also introduces a signal that can be exploited for membership inference. In this work, we address this issue by proposing a federated Gaussian Process learning framework based on key-dependent orthogonal feature transformations.

Motivated by the need for inference-time privacy robustness in federated Gaussian Process models, we design a unified framework that combines scalable RFF-based learning with key-dependent orthogonal encryption and multi-key authorization. The main contributions of this study are as follows:

- We propose a secure computation framework for Random Fourier Feature-based Gaussian Process Regression using key-dependent orthogonal feature encryption. We theoretically show that predictive mean and variance remain identical to the plaintext model when the correct key is used, while the model remains protected.
- We extend the proposed secure RFF-GPR framework to federated learning using ADMM-based distributed optimization. The proposed method enables privacy preserving distributed training while maintaining predictive consistency under correct key usage
- We introduce a multi-key access control mechanism in federated Gaussian Process regression. We show

that predictive variance increases under key mismatch, which reduces the effectiveness of variance-based membership inference attacks while preserving prediction accuracy for authorized users.

The remainder of this study is organized as follows: Section II reviews an overview of the related work. Section III presents the proposed secure Random Fourier Feature–based GPR framework, including the orthogonal encryption mechanism and its theoretical properties. Section IV extends this framework to the federated setting and introduces the proposed multi-key access control mechanism with ADMM-based distributed training. Section V reports the experimental results and evaluates robustness, privacy behavior and multi-key access performance. Section VI presents the discussion and limitations of the proposed framework. Finally, Section VII concludes the study and outlines directions for future work.

II. Related Work

A. Membership Inference Attack

Membership inference attacks (MIAs) aim to determine whether a specific data record was included in a model's training set by analyzing the model's output behavior [14], [15], [16]. Such attacks exploit systematic differences between training and non-training samples, including variations in confidence scores or uncertainty estimates. Fig. 1 provides a conceptual overview of an MIA's architecture. The diagram shows how an adversary trains an attack model to differentiate between member and non-member data points based on observed prediction outputs after using a shadow model to imitate the behavior of a target model.

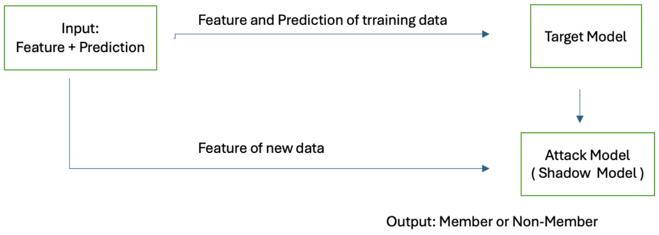

Fig. 1. Architecture of a Membership Inference Attack (MIA) framework.

GPR models are particularly relevant in this context because they explicitly output predictive variance. In standard deployments, predictive variance is typically lower for inputs close to the training data and higher for unseen inputs. This variance gap can serve as a signal for membership inference, creating privacy risks even when raw data are not shared. Therefore, inference-time behavior of predictive uncertainty becomes a critical consideration in privacy-preserving GPR.

B. Federated Learning

FL is a decentralized paradigm that enables collaborative model training across distributed data sources without sharing raw data [17]. Instead of transferring datasets, clients compute local updates that are aggregated by a central server. This paradigm reduces risks associated with centralized data collection and supports regulatory compliance in privacy-sensitive domains. FL is closely connected to distributed optimization and privacy-preserving learning frameworks, including ADMM-based approaches, which offer principled strategies for coordinating local updates and enhancing communication efficiency in decentralized settings [18].

C. Secure Computation and Privacy-Preserving Gaussian Process

Privacy-preserving machine learning has also been investigated through secure computation and cryptographic methods, such as secure multi-party protocols for linear regression and classification [19]. While these schemes offer strong formal privacy guarantees, they typically introduce significant computational costs and can be impractical for federated inference scenarios involving frequent queries. For Gaussian Process models, Random Unitary Transformations (RUT) and related orthogonal transformations have been proposed to conceal feature representations while preserving geometric properties required for kernel evaluation [20], [21], [22], [23], [24]. Selective feature encryption further applies such transformations only to user-defined sensitive attributes [25], improving computational efficiency compared to full encryption.

Prior approaches to privacy-preserving Gaussian Process models and federated learning primarily assume centralized deployments or focus on training-time privacy mechanisms such as differential privacy, homomorphic encryption, or secure aggregation. While these methods provide certain privacy guarantees, they either introduce accuracy degradation, incur significant computational overhead, or do not address inference-time privacy risks. In particular, existing approaches do not explicitly consider inference-time access control or the role of predictive uncertainty in mitigating membership inference attacks in federated GPR settings. In contrast, the proposed framework integrates orthogonal feature transformation with federated RFF-based GPR and introduces a key-dependent access control mechanism. This enables consistent prediction performance for authorized users while intentionally degrading prediction reliability under key mismatch. Furthermore, predictive variance is leveraged as a mechanism for enforcing access control, which distinguishes the proposed method from existing privacy-preserving approaches.

III. Proposed Secure Random Fourier Feature-Based Gaussian Process Regression

A. Gaussian Process Regression Formulation

We focus on a regression task, where the inputs are feature vectors:

$\mathbf{x}_i \in \mathbb{R}^D$

and the corresponding outputs are scalar target values:

$y_i \in \mathbb{R}.$

Suppose we have a training dataset:

$\mathcal{D}_{\text{train}} = \{X, Y\},$

where, the input and output matrices are represented as in Eq. (1),

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}. \tag{1}$$

The outputs are modeled as the sum of an underlying latent function and Gaussian noise, as shown in Eq. (2):

$$\mathbf{Y} = f(\mathbf{X}) + \epsilon, \tag{2}$$

where, $f(\mathbf{X})$ denotes the unknown latent function to be inferred, and $\epsilon \sim \mathcal{N}(0, \sigma^2 I_N)$ represents independent Gaussian noise with zero mean and variance σ^2. This noise term captures uncertainty and possible measurement errors in the observations. The latent function $f(\mathbf{X})$ is assumed to follow a Gaussian Process (GP) prior, as expressed in Eq. (3):

$$f(\mathbf{X}) \sim \mathcal{GP}(0, \mathbf{K}(\mathbf{X}, \mathbf{X})), \tag{3}$$

where, the mean is set to zero for simplicity, and $\mathbf{K}(\mathbf{X}, \mathbf{X})$ is the covariance matrix computed using a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$. The kernel defines the similarity between two input points and controls the smoothness and complexity of the underlying function. A commonly used kernel is the Radial Basis Function (RBF), or squared exponential kernel, defined as in Eq. (4):

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\theta_2}\right) + \sigma^2 \delta(i, j), \tag{4}$$

where, θ_1 and θ_2 are kernel hyperparameters representing the signal variance and length scale respectively, and $\delta(i, j) = 1$ if $i = j$ and 0 otherwise. The first term models smooth correlations between input points, while the second term accounts for independent noise. The hyperparameters $\{\theta_1, \theta_2, \sigma^2\}$ are estimated by minimizing the negative log marginal likelihood (NLML) given in Eq. (5):

$$\begin{aligned}\mathcal{L} = \tfrac{1}{2}\mathbf{Y}^\top\big(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 I\big)^{-1}\mathbf{Y} + \tfrac{1}{2}\log\big|\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I}\big| \\ + \tfrac{N}{2}\log(2\pi).\end{aligned} \tag{5}$$

This loss function balances model fit (the first term) and model complexity (the second term). Optimization is typically performed via gradient-based methods such as conjugate gradient or L-BFGS.

Once the model is trained, the predictive distribution for a new test input $\mathbf{x}_*$ is Gaussian, with closed-form expressions for the predictive mean and variance as:

$$\mu(\mathbf{x}_*) = \mathbf{K}(\mathbf{x}_*, \mathbf{X})[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I}]^{-1}\mathbf{Y}, \tag{6}$$

$$\mathrm{Var}(\mathbf{x}_*) = \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{K}(\mathbf{x}_*, \mathbf{X})[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I}]^{-1}\mathbf{K}(\mathbf{X}, \mathbf{x}_*). \tag{7}$$

Eq. (6) provides the expected prediction at $\mathbf{x}_*$, while Eq. (7) quantifies the model's uncertainty in that prediction. GPR thus offers both mean estimates and confidence intervals, making it especially useful for uncertainty-aware modeling. However, the computation of the inverse covariance matrix $[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I}]^{-1}$ scales as $\mathcal{O}(N^3)$, which becomes intractable for large datasets.

B. Random Fourier Feature Approximation of the Kernel

Although Gaussian Processes (GPs) provide excellent predictive performance and uncertainty estimation, their computational complexity scales cubically with the number of training samples N because of the matrix inversion in the covariance term $[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I}]^{-1}$. To address this limitation, Random Fourier Features (RFF) are introduced as a kernel approximation technique that transforms the nonparametric GP model into a computationally efficient linear model in a randomized feature space.

According to Bochner's theorem, any continuous, shift-invariant kernel,

$$k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i - \mathbf{x}_j)$$

can be represented as the Fourier transform of a probability density function $p(\boldsymbol{\omega})$, as shown in Eq. (8):

$$k(\mathbf{x}_i, \mathbf{x}_j) = \int_{\mathbb{R}^D} p(\boldsymbol{\omega})\, e^{j\boldsymbol{\omega}^\top(\mathbf{x}_i - \mathbf{x}_j)}\, d\boldsymbol{\omega}. \tag{8}$$

For the Radial Basis Function (RBF) kernel, the corresponding spectral density $p(\boldsymbol{\omega})$ is Gaussian, that is $\boldsymbol{\omega} \sim \mathcal{N}(0, \Sigma)$, where Σ controls the length-scale parameter of the kernel. Using Monte Carlo sampling, the integral in (9) can be approximated by drawing m samples $\{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_m\}$ from $p(\boldsymbol{\omega})$ and constructing a low-dimensional feature mapping $\boldsymbol{\phi}(\mathbf{x})$ as Eq. (9):

$$\boldsymbol{\phi}(\mathbf{x}) = \sqrt{\frac{2}{m}} \begin{bmatrix} \cos(\boldsymbol{\omega}_1^\top \mathbf{x} + b_1) \\ \cos(\boldsymbol{\omega}_2^\top \mathbf{x} + b_2) \\ \vdots \\ \cos(\boldsymbol{\omega}_m^\top \mathbf{x} + b_m) \end{bmatrix}, \tag{9}$$

where, $b_i \sim \mathrm{Uniform}[0, 2\pi)$ are randomly sampled phase shifts. With this mapping, the original kernel function in Eq. (10) can be approximated as an inner product in the transformed feature space:

$$k(\mathbf{x}_i, \mathbf{x}_j) \approx \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_j). \tag{10}$$

This transformation enables the GP model to be rewritten in a parametric form that depends only on the feature dimension m (typically $m \ll N$), dramatically reducing computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(Nm^2)$.

After applying the RFF mapping to all input samples, the dataset becomes $\Phi = \boldsymbol{\phi}(X) \in \mathbb{R}^{N \times m}$. The regression model can then be expressed, as shown in Eq. (11):

$$\mathbf{Y} = \Phi\mathbf{w} + \epsilon, \tag{11}$$

where, $\mathbf{w} \in \mathbb{R}^m$ represents the linear weights in the RFF space, and $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ is Gaussian noise. The optimal weight vector $\mathbf{w}$ is obtained by minimizing the ridge regression objective given in Eq. (12):

$$\min_{\mathbf{w}} \frac{1}{2}\|\Phi\mathbf{w} - Y\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2, \tag{12}$$

whose closed-form solution is given by:

$$\mathbf{w}^* = (\Phi^\top\Phi + \lambda I_m)^{-1}\Phi^\top Y. \tag{13}$$

While Eq. (13) provides the closed-form solution for the ridge-regression weights, GPR additionally requires the predictive mean and predictive variance. Under the Random Fourier Feature (RFF) approximation, the GP model can be interpreted as Bayesian linear regression in the randomized feature space. Let $\Phi = \phi(X) \in \mathbb{R}^{N\times m}$ denote the RFF-transformed training feature matrix, as defined in Eq. (14) and assume:

$$\mathbf{Y} = \Phi\mathbf{w} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 I). \tag{14}$$

With a Gaussian prior $\mathbf{w} \sim \mathcal{N}(0, \lambda^{-1}I)$, the posterior distribution is given in Eq. (15):

$$\mathbf{w} \mid \mathcal{D} \sim \mathcal{N}\left((\Phi^\top\Phi + \lambda I)^{-1}\Phi^\top\mathbf{Y},\ \sigma^2(\Phi^\top\Phi + \lambda I)^{-1}\right). \tag{15}$$

For a test input $\mathbf{x}_*$ with RFF representation $\boldsymbol{\phi}_* = \phi(\mathbf{x}_*)$, the predictive mean and variance are given in Eq. (16) and Eq. (17), respectively:

$$\mu_{\mathrm{RFF}}(\mathbf{x}_*) = \boldsymbol{\phi}_*^\top(\Phi^\top\Phi + \lambda I)^{-1}\Phi^\top\mathbf{Y}, \tag{16}$$

$$\mathrm{Var}_{\mathrm{RFF}}(\mathbf{x}_*) = \sigma^2 + \sigma^2\,\boldsymbol{\phi}_*^\top(\Phi^\top\Phi + \lambda I)^{-1}\boldsymbol{\phi}_*. \tag{17}$$

C. Orthogonal Encryption Theory and Secure Computation

To protect input representations while preserving Gaussian Process behavior, we introduce a key-dependent orthogonal transformation in the input space and formally establish its invariance properties.

1) Orthogonal transformation and distance preservation: Let

$$\boldsymbol{Q} \in \mathbb{R}^{D\times D}$$

be a random orthogonal matrix satisfying [see Eq. (18)]:

$$\boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^\top = I. \tag{18}$$

For any two input vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$, define their encrypted forms as Eq. (19):

$$\mathbf{x}_i^{(e)} = \boldsymbol{Q}\mathbf{x}_i, \quad \mathbf{x}_j^{(e)} = \boldsymbol{Q}\mathbf{x}_j. \tag{19}$$

The squared Euclidean distance between encrypted inputs is Eq. (20):

$$\begin{aligned}\|\mathbf{x}_i^{(e)} - \mathbf{x}_j^{(e)}\|^2 &= \|\boldsymbol{Q}(\mathbf{x}_i - \mathbf{x}_j)\|^2 \\ &= (\mathbf{x}_i - \mathbf{x}_j)^\top\boldsymbol{Q}^\top\boldsymbol{Q}(\mathbf{x}_i - \mathbf{x}_j).\end{aligned} \tag{20}$$

Using $\boldsymbol{Q}^\top\boldsymbol{Q} = I$, we obtain Eq. (21):

$$\|\mathbf{x}_i^{(e)} - \mathbf{x}_j^{(e)}\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^\top(\mathbf{x}_i - \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2. \tag{21}$$

Thus, Euclidean distances are preserved under orthogonal transformation.

2) Invariance of RFF projection: In the Random Fourier Feature (RFF) approximation, feature mappings depend on inner products of the form [see Eq. (22)]:

$$\boldsymbol{\omega}^\top\boldsymbol{x}. \tag{22}$$

After encryption [see Eq. (23)],

$$\boldsymbol{x}^{(e)} = \boldsymbol{Q}\boldsymbol{x}. \tag{23}$$

To preserve projection consistency, we rotate the spectral frequencies as Eq. (24):

$$\boldsymbol{\omega}^{(e)} = \boldsymbol{Q}^\top\boldsymbol{\omega}. \tag{24}$$

The encrypted projection becomes Eq. (25):

$$\begin{aligned}(\boldsymbol{\omega}^{(e)})^\top\boldsymbol{x}^{(e)} &= (\boldsymbol{Q}^\top\boldsymbol{\omega})^\top(\boldsymbol{Q}\boldsymbol{x}) \\ &= \boldsymbol{\omega}^\top\boldsymbol{Q}\boldsymbol{Q}^\top\boldsymbol{x}.\end{aligned} \tag{25}$$

Since $\boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{I}$, we obtain Eq. (26):

$$(\boldsymbol{\omega}^{(e)})^\top\boldsymbol{x}^{(e)} = \boldsymbol{\omega}^\top\boldsymbol{x}. \tag{26}$$

3) Invariance of feature mapping: The RFF mapping is defined as Eq. (27):

$$\phi(\boldsymbol{x}) = \sqrt{\frac{2}{m}}\cos(\boldsymbol{\Omega}^\top\boldsymbol{x} + \boldsymbol{b}). \tag{27}$$

Under encryption [see Eq. (28)],

$$\phi^{(e)}(\boldsymbol{x}) = \sqrt{\frac{2}{m}}\cos\Big((\boldsymbol{\Omega}^{(e)})^\top\boldsymbol{x}^{(e)} + \boldsymbol{b}\Big). \tag{28}$$

Using the projection invariance result [see Eq. (29)],

$$\phi^{(e)}(\boldsymbol{x}) = \phi(\boldsymbol{x}). \tag{29}$$

Thus, the encrypted feature matrix satisfies [see Eq. (30)]:

$$\boldsymbol{\Phi}^{(e)} = \boldsymbol{\Phi}. \tag{30}$$

4) Preservation of predictive mean and variance: Since $\boldsymbol{\Phi}^{(e)} = \boldsymbol{\Phi}$, we have Eq. (31):

$$\boldsymbol{\Phi}^{(e)\top}\boldsymbol{\Phi}^{(e)} = \boldsymbol{\Phi}^\top\boldsymbol{\Phi}. \tag{31}$$

The ridge-regression solution, therefore, satisfies [see Eq. (32)]:

$$\boldsymbol{w}^{*(e)} = \left(\boldsymbol{\Phi}^{(e)\top}\boldsymbol{\Phi}^{(e)} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}^{(e)\top}\boldsymbol{y} = \boldsymbol{w}^*. \tag{32}$$

For any test input $\boldsymbol{x}^*$ [see Eq. (33) and Eq. (34)]:

$$\mu_{\text{enc}}(\boldsymbol{x}^*) = \mu_{\text{plain}}(\boldsymbol{x}^*), \tag{33}$$

$$\text{Var}_{\text{enc}}(\boldsymbol{x}^*) = \text{Var}_{\text{plain}}(\boldsymbol{x}^*). \tag{34}$$

Thus, orthogonal encryption preserves both predictive accuracy and uncertainty exactly under consistent key usage.

IV. Proposed Federated Secure Gaussian Process Regression with Multi-key Access Control

A. System Overview

Before presenting the detailed ADMM formulation and key-based access control mechanism, we first describe the overall system architecture of the proposed federated secure Gaussian Process regression framework.

The system consists of K distributed clients and a central aggregation server operating under a horizontal federated learning setting. Each client $\mathcal{C}_k$ holds its own private dataset $\mathcal{D}_k = \{\boldsymbol{X}_k, \boldsymbol{y}_k\}$, and raw data never leave the client side. The overall workflow proceeds as follows:

- Local Feature Transformation: Each client applies the Random Fourier Feature (RFF) mapping to its local data to obtain Eq. (35):

$$\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(\boldsymbol{X}_k). \tag{35}$$

If encryption is enabled, the client applies a key-dependent orthogonal transformation to both the input features and the spectral frequencies before computing the RFF representation, ensuring representation consistency under correct key usage.

- Local Sufficient Statistics Computation: Instead of transmitting raw data, each client computes the local quantities [see Eq. (36)].

$$\boldsymbol{\Phi}_k^\top\boldsymbol{\Phi}_k \quad \text{and} \quad \boldsymbol{\Phi}_k^\top\boldsymbol{y}_k \tag{36}$$

which are sufficient for ridge-regression–based RFF Gaussian Process training.

- Federated ADMM Optimization: Each client maintains a local parameter vector w_k and a scaled dual variable u_k. Only parameter vectors w_k and dual variables are transmitted to the server. The central server updates the global consensus variable z, which represents the shared model parameter.
- Global Model for Inference: After convergence, the final predictive model uses the consensus parameter z, and predictions are computed as Eq. (37):

$$\mu(\boldsymbol{x}^*) = \phi(\boldsymbol{x}^*)^\top\boldsymbol{z}. \tag{37}$$

Fig. 2 shows the system architecture of the proposed federated secure Gaussian Process regression framework with multi-key orthogonal feature encryption.

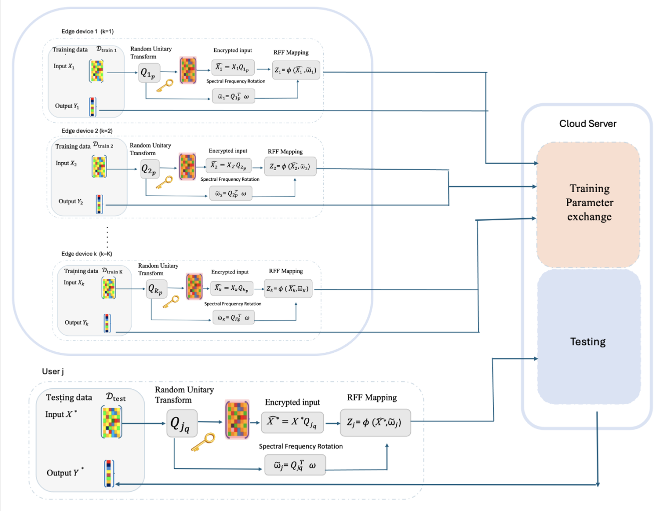
Edge device 1 (k=1)
Training data $\mathcal{D}_{\text{train }1}$
Input X_1
Output Y_1
Random Unitary Transform
Q_{1_p}
Encrypted input
$\widehat{X_1} = X_1 Q_{1_p}$
RFF Mapping
$Z_1 = \phi(\widehat{X_1}, \tilde{\omega}_1)$
Spectral Frequency Rotation
$\tilde{\omega}_1 = Q_{1p}^T\ \omega$
Edge device 2 (k=2)
Training data $\mathcal{D}_{\text{train }2}$
Input X_2
Output Y_2
Random Unitary Transform
Q_{2_p}
Encrypted input
$\widehat{X_2} = X_2 Q_{2_p}$
RFF Mapping
$Z_2 = \phi(\widehat{X_2}, \tilde{\omega}_2)$
Spectral Frequency Rotation
$\tilde{\omega}_2 = Q_{2p}^T\ \omega$
Edge device k (k=K)
Training data $\mathcal{D}_{\text{train K}}$
Input X_k
Output Y_k
Random Unitary Transform
Q_{k_p}
Encrypted input
$\widehat{X_k} = X_k Q_{k_p}$
RFF Mapping
$Z_k = \phi(\widehat{X_k}, \tilde{\omega}_K)$
Spectral Frequency Rotation
$\tilde{\omega}_K = Q_{Kp}^T\ \omega$
User j
Testing data $\mathcal{D}_{\text{test}}$
Input X^*
Output Y^*
Random Unitary Transform
Q_{j_q}
Encrypted input
$\widehat{X^*} = X^* Q_{j_q}$
RFF Mapping
$Z_j = \phi(\widehat{X^*}, \tilde{\omega}_j)$
Spectral Frequency Rotation
$\tilde{\omega}_j = Q_{jq}^T\ \omega$
Cloud Server
Training Parameter exchange
Testing

Fig. 2. Architecture of a federated secure Gaussian process regression framework with multi-key orthogonal feature encryption.

We now extend the secure RFF-based Gaussian Process framework introduced in Section III to the federated learning setting. This extension constitutes the second contribution of this work: a federated secure GPR system with key-dependent access control and multi-key robustness.

B. Federated ADMM-Based Training for RFF Gaussian Process Regression

In the FL setting, data are distributed across K clients $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$. Each client k holds a local dataset $\mathcal{D}_k = \{\boldsymbol{X}_k, \boldsymbol{y}_k\}$. Raw data are not shared with the central server. Instead, clients collaboratively train a global model using the ADMM. After applying the Random Fourier Feature (RFF) mapping, each client constructs a local feature matrix, as defined in Eq. (38):

$$\boldsymbol{\Phi}_k = \boldsymbol{\phi}(\boldsymbol{X}_k) \in \mathbb{R}^{N_k \times m}. \tag{38}$$

The global model is parameterized by a shared weight vector $\boldsymbol{z} \in \mathbb{R}^m$. Each client maintains a local variable $\boldsymbol{w}_k$, with the consensus constraint given in Eq. (39):

$$\boldsymbol{w}_k = \boldsymbol{z}, \quad \forall k. \tag{39}$$

1) Global optimization problem: The federated RFF-based regression objective implemented in this work is given in Eq. (40):

$$\begin{aligned} \min_{\{\boldsymbol{w}_k\},\, \boldsymbol{z}} \quad & \sum_{k=1}^{K} \frac{1}{2}\|\boldsymbol{\Phi}_k \boldsymbol{w}_k - \boldsymbol{y}_k\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{z}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{w}_k = \boldsymbol{z}, \quad \forall k. \end{aligned} \tag{40}$$

Regularization is applied to the global consensus variable $\boldsymbol{z}$, ensuring that the final model is governed by a single shared parameter vector.

2) Scaled ADMM formulation: We adopt the scaled form of ADMM. Let $\boldsymbol{u}_k$ denote the scaled dual variable for client k. The augmented Lagrangian is defined in Eq. (41):

$$\mathcal{L}_\rho = \sum_{k=1}^{K} \left[\frac{1}{2}\|\boldsymbol{\Phi}_k \boldsymbol{w}_k - \boldsymbol{y}_k\|_2^2 + \frac{\rho}{2}\|\boldsymbol{w}_k - \boldsymbol{z} + \boldsymbol{u}_k\|_2^2 \right] + \frac{\lambda}{2}\|\boldsymbol{z}\|_2^2. \tag{41}$$

3) ADMM iterative updates: Each ADMM iteration consists of the following three steps:

a) Local client update: Each client solves Eq. (42):

$$\boldsymbol{w}_k^{(t+1)} = \arg\min_{\boldsymbol{w}_k} \left[\frac{1}{2}\|\boldsymbol{\Phi}_k \boldsymbol{w}_k - \boldsymbol{y}_k\|_2^2 + \frac{\rho}{2}\|\boldsymbol{w}_k - \boldsymbol{z}^{(t)} + \boldsymbol{u}_k^{(t)}\|_2^2 \right]. \tag{42}$$

This yields the closed-form solution, Eq. (43):

$$\boldsymbol{w}_k^{(t+1)} = (\boldsymbol{\Phi}_k^\top \boldsymbol{\Phi}_k + \rho \boldsymbol{I}_m)^{-1} \left(\boldsymbol{\Phi}_k^\top \boldsymbol{y}_k + \rho(\boldsymbol{z}^{(t)} - \boldsymbol{u}_k^{(t)}) \right). \tag{43}$$

This update is performed independently on each client without sharing local data.

b) Global consensus update: After receiving $\boldsymbol{w}_k^{(t+1)}$ from all clients, the server computes, Eq. (44):

$$\boldsymbol{z}^{(t+1)} = \left(K\boldsymbol{I}_m + \frac{\lambda}{\rho}\boldsymbol{I}_m \right)^{-1} \sum_{k=1}^{K} \left(\boldsymbol{w}_k^{(t+1)} + \boldsymbol{u}_k^{(t)} \right). \tag{44}$$

Equivalently [see Eq. (45)],

$$\boldsymbol{z}^{(t+1)} = \frac{1}{K + \lambda/\rho} \sum_{k=1}^{K} \left(\boldsymbol{w}_k^{(t+1)} + \boldsymbol{u}_k^{(t)} \right). \tag{45}$$

This step enforces global consensus and incorporates ridge regularization.

c) Dual variable update: Each client updates its scaled dual variable [see Eq. (46)]:

$$\boldsymbol{u}_k^{(t+1)} = \boldsymbol{u}_k^{(t)} + \boldsymbol{w}_k^{(t+1)} - \boldsymbol{z}^{(t+1)}. \tag{46}$$

After convergence, the global parameter vector $\boldsymbol{z}$ is used for prediction [see Eq. (47)]:

$$\mu(\boldsymbol{x}^*) = \phi(\boldsymbol{x}^*)^\top \boldsymbol{z}. \tag{47}$$

Thus, federated training preserves the RFF-based structure while enabling decentralized optimization without raw data sharing.

C. Encryption Mechanism in Federated Setting

Let k_p and j_q denote the indices of the private keys used for encryption, where each key corresponds to a specific client or access privilege. The notation $\boldsymbol{Q}_{k_p}$ represents the orthogonal transformation applied by client k using private key p. For notational simplicity, we hereafter denote the private key indices associated with encryption and inference as p and q, respectively, while omitting the client subscript when no ambiguity arises.

Let client k hold a local input matrix $\boldsymbol{X}_k \in \mathbb{R}^{N_k \times D}$. A client-specific random orthogonal matrix $\boldsymbol{Q}_{k_p} \in \mathbb{R}^{D \times D}$ is generated such that it satisfies the orthogonality condition in Eq. (48):

$$\boldsymbol{Q}_{\boldsymbol{k_p}}^\top \boldsymbol{Q}_{j_q} = \boldsymbol{I}_D. \tag{48}$$

Let $k \in \{1, 2, 3, \ldots, K\}$ denote the client index. The encrypted input is defined as in Eq. (49):

$$\widetilde{\boldsymbol{X}}_k = \boldsymbol{X}_k \boldsymbol{Q}_{k_p}. \tag{49}$$

Under the RFF approximation of the RBF kernel, the feature mapping is in Eq. (50):

$$\boldsymbol{\phi}(\boldsymbol{x}) = \sqrt{\frac{2}{m}} \cos\!\left(\boldsymbol{\Omega}^\top \boldsymbol{x} + \boldsymbol{b}\right), \tag{50}$$

where,

$\boldsymbol{\Omega} \in \mathbb{R}^{D \times m}$ contains Gaussian random frequencies and $\boldsymbol{b}$ is a random phase vector. To preserve consistency inside the cosine nonlinearity, the spectral frequencies are rotated as in Eq. (51):

$$\boldsymbol{\Omega}_k = \boldsymbol{Q}_{k_p}^\top \boldsymbol{\Omega}. \tag{51}$$

The encrypted RFF representation is defined in Eq. (52):

$$\boldsymbol{\Phi}_k^{(e)} = \sqrt{\frac{2}{m}} \cos\Big(\widetilde{\boldsymbol{X}}_k \boldsymbol{\Omega}_k + \boldsymbol{b}\Big). \tag{52}$$

Substituting [see Eq. (53)]:

$\widetilde{\boldsymbol{X}}_k = \boldsymbol{X}_k \boldsymbol{Q}_{k_p}$

and

$\boldsymbol{\Omega}_k = \boldsymbol{Q}_{k_p}^{\top} \boldsymbol{\Omega}$,

$$\begin{aligned} \boldsymbol{\Phi}_k^{(e)} &= \sqrt{\frac{2}{m}} \cos\Big(\boldsymbol{X}_k \boldsymbol{Q}_{k_p} \boldsymbol{Q}_{k_p}^{\top} \boldsymbol{\Omega} + \boldsymbol{b}\Big) \\ &= \sqrt{\frac{2}{m}} \cos(\boldsymbol{X}_k \boldsymbol{\Omega} + \boldsymbol{b}) \\ &= \boldsymbol{\Phi}_k. \end{aligned} \tag{53}$$

For equality with the plaintext representation, we would require [see Eq. (54)]:

$$\boldsymbol{Q}_{k_p}^{T} \boldsymbol{Q}_{j_q} = \boldsymbol{I} \quad \Longrightarrow \quad \boldsymbol{Q}_{k_p} = \boldsymbol{Q}_{j_q}. \tag{54}$$

When $\boldsymbol{Q}_{k_p} \neq \boldsymbol{Q}_{k_q}$, the feature representations no longer align. Consequently, the predictive mean deviates and the predictive variance typically becomes uniformly large. This intentional misalignment forms the mathematical basis for key-based access control.

D. Key-Based Access Control for Client-Level Metrics

Beyond cryptographic alignment, the framework restricts information exposure at the system interface level. Let K clients be indexed by $\mathcal{K} = \{1, \ldots, K\}$, and let a user possess key set $\mathcal{S} \subseteq \mathcal{K}$. We define an authorization function in Eq. (55):

$$\mathsf{Auth}(\mathcal{S}) = \mathcal{S}. \tag{55}$$

For each experimental scenario s, the system computes client-level metrics, as defined in Eq. (56) and Eq. (57):

$$\mathbf{m}_{k,s} = \big(\mathrm{MAE}_{k,s}, \mathrm{AvgVar}_{k,s}\big). \tag{56}$$

The user is permitted to observe only

$$\{\mathbf{m}_{k,s} \mid k \in \mathsf{Auth}(\mathcal{S})\}. \tag{57}$$

No raw predictions, no per-sample variances, and no global aggregates are disclosed. This metrics-only policy further reduces the attack surface for strong MIAs.

Encrypted input processing remains fully compatible with federated ADMM optimization. Each client performs local updates using its encrypted RFF representation, while the server coordinates consensus variables. Importantly, encryption does not alter the ADMM objective, does not increase communication complexity, and does not modify optimization dynamics.

E. Security and Efficiency

Orthogonal encryption introduces minimal computational overhead, requiring only matrix multiplications of complexity $\mathcal{O}(N_k D)$ per client. Because the RFF feature matrix remains algebraically identical under same-key encryption, predictive accuracy and uncertainty calibration are preserved exactly. Overall, the proposed framework integrates securely and efficiently with federated RFF-based GPR, providing key-based inference-time access control while maintaining full model fidelity.

V. Experimental Results

A. Experimental Setup

Experiments were conducted using the publicly available NHANES 2021–2023 (L cycle) dataset. After preprocessing and restricting to adults (age $\geq$ 18 years), the final dataset contains $N = 5277$ samples with $D = 5$ input features (age, sex, ethnicity, income-to-poverty ratio, and body mass index) [26]. The regression target is the mean systolic blood pressure (SBP), computed from repeated SBP measurements after removing invalid entries. All input features were standardized using z-score normalization.

The input matrix is denoted as $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ and the target vector as $\boldsymbol{Y} \in \mathbb{R}^{N}$. A horizontal FL setting with $K = 5$ clients was simulated by randomly partitioning the dataset into five disjoint subsets (approximately 1055–1056 samples per client). Each client performed an 80/20 local train–test split, yielding approximately 844 training samples and 211–212 testing samples per client. No raw data were shared during training.

GPR was approximated using Random Fourier Features (RFF) with an RBF kernel and $m = 300$ random features, resulting in a linear ridge regression formulation in feature space. Federated optimization was performed using ADMM with regularization parameter $\lambda = 10^{-3}$, penalty parameter $\rho = 1.0$, and 20 iterations. The predictive noise variance was estimated from training residuals.

For secure computation, each client applied a RUT to its local input data. The orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{5 \times 5}$ was generated via QR decomposition of a random Gaussian matrix. Distinct keys were assigned to different clients, and both key-consistent and key-mismatch scenarios were evaluated.

TABLE I. Conventional Method: Training and Testing Data are Not Encrypted

(a) Test Included			(b) Test Excluded		
Client	**Average Error**	**Average Variance**	**Client**	**Average Error**	**Average Variance**
1	12.118	62.563	1	13.046	270.649
2	12.739	64.633	2	13.533	277.650
3	12.144	67.212	3	12.889	342.483
4	11.443	64.092	4	12.024	214.247
5	13.214	66.843	5	14.132	358.765

B. Plaintext Federated Gaussian Process Baseline

We now analyze the plaintext federated Gaussian Process model, where neither training nor testing data are encrypted. All numerical results for these baseline conditions are summarized in Table I, while the corresponding behaviors are illustrated in Fig. 1 and Fig. 2.

TABLE II. PROPOSED CASE 1: TRAINING AND TESTING DATA ARE ENCRYPTED

(a) Test Included ($p = q$)			(b) Test Excluded ($p = q$)		
Client	**Average Error**	**Average Variance**	**Client**	**Average Error**	**Average Variance**
1	12.118	62.563	1	13.046	270.649
2	12.739	64.633	2	13.533	277.650
3	12.144	67.212	3	12.889	342.483
4	11.443	64.092	4	12.024	214.247
5	13.214	66.843	5	14.132	358.765
(a) Test Included ($p \neq q$)			**(b) Test Excluded ($p \neq q$)**		
Client	**Average Error**	**Average Variance**	**Client**	**Average Error**	**Average Variance**
1	37.223	6752.542	1	36.452	8596.941
2	28.912	6938.149	2	28.253	8474.300
3	34.213	6420.077	3	34.104	8318.913
4	29.628	6326.077	4	29.327	7686.162
5	29.329	5392.608	5	28.558	6961.341

TABLE III. PROPOSED CASE 2: ONLY TRAINING DATA IS ENCRYPTED, TESTING DATA IS NOT ENCRYPTED

(a) Test Included			(b) Test Excluded		
Client	**Average Error**	**Average Variance**	**Client**	**Average Error**	**Average Variance**
1	31.665	4851.219	1	31.686	6184.626
2	27.153	4067.699	2	26.074	5009.210
3	26.935	6188.178	3	26.351	7849.094
4	28.273	6505.624	4	27.372	7768.557
5	24.816	2570.037	5	24.920	3386.514

1) Test data included in training: As illustrated in Fig. 3, when test samples are included in the training set, the model achieves low prediction error and low predictive variance across all clients. For Client 1, the average error and variance are 12.118 and 62.563, respectively. As reported in Table I, similar results are observed for Clients 2–5, with average errors ranging from 11.443 to 13.214 and predictive variances remaining around 64–67. This behavior reflects strong data fitting and high predictive confidence, as expected when Gaussian Process models are evaluated on inputs that were already observed during training.

2) Test data excluded from training: In contrast, Fig. 4 shows the case where test samples are excluded from the training set. For Client 1, the average error increases modestly to 13.046, while the predictive variance rises substantially to 270.649. As summarized in Table I, the same pattern is observed for Clients 2–5, where average errors increase slightly (13.533, 12.889, 12.024, and 14.132), but predictive variances grow significantly (277.650, 342.483, 214.247, and 358.765). This increase in variance indicates appropriate uncertainty calibration for unseen inputs. However, the clear variance gap between training and non-training samples also highlights a potential privacy concern, as such differences may be exploited to infer membership information in conventional Gaussian Process models.

TABLE IV. DIFFERENTIAL PRIVACY GAUSSIAN PROCESS CASE

Condition	Average Error	Average Variance
Test Data Included	25.855	64.574
Test Data Excluded	28.468	64.646

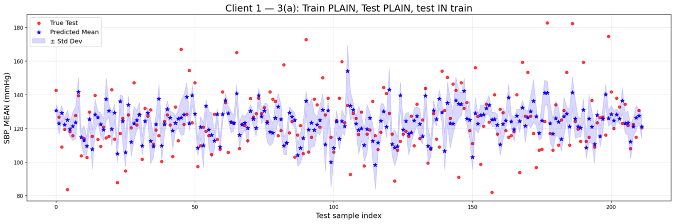

Fig. 3. Mean prediction and uncertainty of conventional GP for non-encrypted training and testing data (Test included in Training).

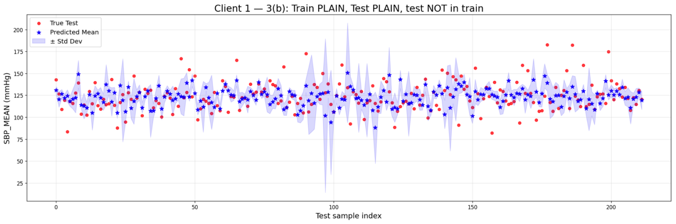

Fig. 4. Mean prediction and uncertainty of conventional GP for non-encrypted training and testing data (Test not included in Training).

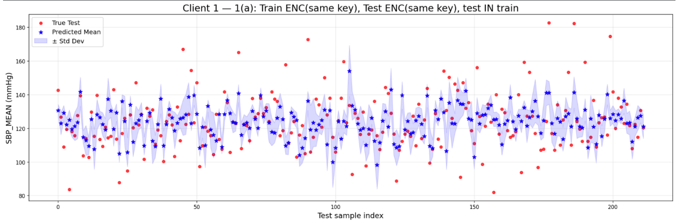

Fig. 5. Mean prediction and uncertainty of proposed method 1 ($\boldsymbol{Q}_k = \boldsymbol{Q}_j$) for non-encrypted training and testing data (Test included in Training).

C. Case 1: Authorized Encrypted Inference

We now analyze the encrypted federated Gaussian Process model under both authorized (same-key) and unauthorized (different-key) settings. All numerical results for these conditions are summarized in Table II, while Fig. 3 to Fig. 6 illustrate the corresponding behaviors.

1) Same-key setting ($p = q$): Under authorized access ($p = q$), the same encryption key is used for both training and testing data. As illustrated in Fig. 5, when test samples are included in the training set, the model maintains low prediction error and low predictive variance. For Client 1, the average error and variance are 12.118 and 62.563, respectively. As reported in Table II, similar results are observed across Clients 2–5, with average errors ranging from 11.443 to 13.214 and predictive variances remaining around 64–67. These values are comparable to the plaintext baseline, indicating that encryption does not degrade performance when the correct key is used.

In Fig. 6, where test samples are excluded from training, the average error increases modestly (e.g., 13.046 for Client 1), while predictive variance rises substantially (270.649 for Client 1). As shown in Table II, the same pattern holds across the remaining clients, with moderate increases in error and large increases in variance (above 200 in all cases). This transition mirrors the plaintext setting, confirming that uncertainty remains properly calibrated under authorized encrypted inference.

2) Different-key setting ($p \neq q$): We next consider the unauthorized scenario ($p \neq q$), where mismatched encryption keys are used for training and testing.

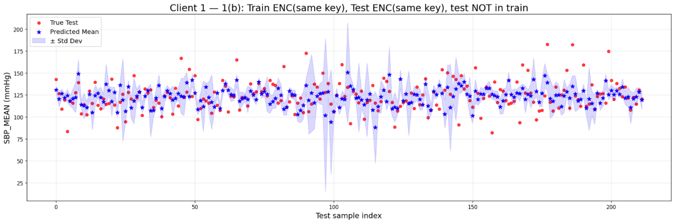

Fig. 6. Mean prediction and uncertainty of proposed method 1 ($\boldsymbol{Q}_k = \boldsymbol{Q}_j$) for non-encrypted training and testing data (Test is not included in Training).

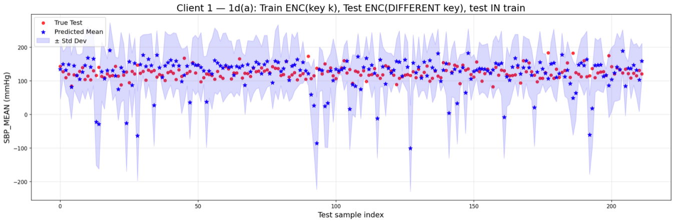

Fig. 7. Mean prediction and uncertainty of proposed method 1 ($\boldsymbol{Q}_k \neq \boldsymbol{Q}_j$) for non-encrypted training and testing data (Test included in Training).

As shown in Fig. 7, when test samples are included in training, predictive performance deteriorates sharply. For Client 1, the average error increases to 37.223 and the variance rises to 6752.542. According to Table II, all clients exhibit similarly large deviations, with errors between approximately 28 and 37 and variances exceeding 5000.

In Fig. 8, where test samples are excluded from training, instability becomes even more pronounced. For Client 1, the average error remains extremely high (36.452) and the predictive variance increases further to 8596.941. As summarized in Table II, the other clients show comparable behavior, with variances ranging from roughly 7000 to 8500.

These substantial increases in both error and uncertainty confirm that when $Q_k \neq Q_j$, the encrypted representations become misaligned and kernel evaluation fails. This demonstrates the access-control property of the proposed framework: without the correct encryption key, meaningful prediction is not possible.

D. Case 2: Unauthorized or Key-Mismatched Inference

We next analyze the mismatched encryption setting where the training data are encrypted but the testing data remain unencrypted. All numerical results for these conditions are summarized in Table III, while the corresponding behaviors are illustrated in the related figures.

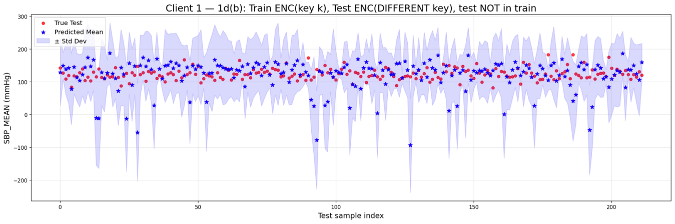

Fig. 8. Mean prediction and uncertainty of proposed method 2 ($\boldsymbol{Q}_k \neq \boldsymbol{Q}_j$) for non-encrypted training and testing data (Test is not included in Training).

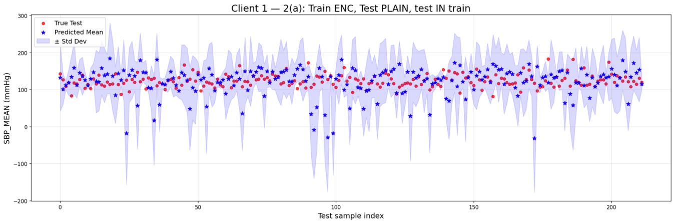

Fig. 9. Mean prediction and uncertainty of proposed method 2 for non-encrypted training and testing data (Test included in Training).

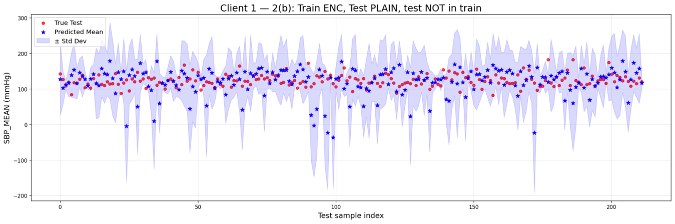

Fig. 10. Mean prediction and uncertainty of proposed method 2 for non-encrypted training and testing data (Test is not included in Training).

1) Mismatched setting: Training encrypted, testing plain (test data included in training): In this configuration, the model is trained on encrypted inputs but evaluated on plaintext test data, even though those test samples are included in the training set, as illustrated in Fig. 9. As reported in Table III, the predictive performance deteriorates substantially across all clients. For Client 1, the average error increases to 31.665 and the predictive variance rises sharply to 4851.219. Similar behavior is observed for Clients 2–5, with average errors ranging from 24.816 to 28.273 and predictive variances between 2570.037 and 6505.624. This mismatch leads to severe degradation in both prediction error and predictive uncertainty. Despite the inclusion of test samples in training, the large increases in both error and variance indicate that the encrypted training representation is incompatible with the unencrypted testing representation. As a result, the kernel evaluations become inconsistent and prediction quality degrades significantly.

2) Mismatched setting: Training encrypted, testing plain (test data excluded from training): When test samples are excluded from training, the instability becomes even more pronounced, as illustrated in Fig. 10. For Client 1, the average error remains high at 31.686, while the predictive variance further increases to 6184.626. As summarized in Table III, the remaining clients show similarly elevated errors (26.074, 26.351, 27.372, and 24.920) and extremely large variances (5009.210, 7849.094, 7768.557, and 3386.514). These results confirm that when the training and testing representations are not aligned, meaningful Gaussian Process inference is no longer possible. The substantial degradation in both predictive accuracy and uncertainty calibration demonstrates the sensitivity of kernel-based learning to representation consistency.

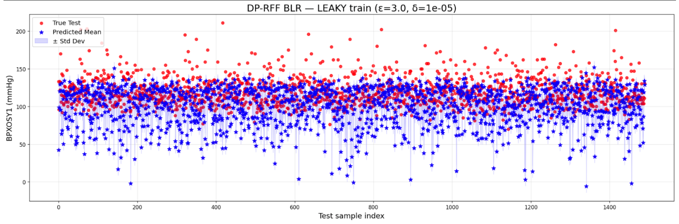

Fig. 11. Mean prediction and uncertainty of DP-GPR (test included in training).

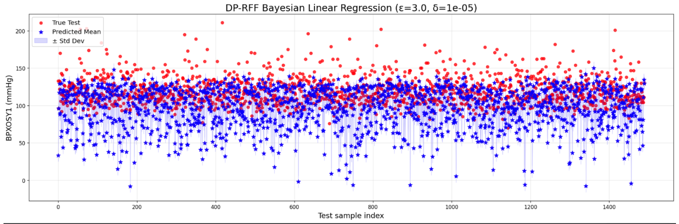

Fig. 12. Mean prediction and uncertainty of DP-GPR (test is not included in training).

E. Differential Privacy Case

We additionally evaluate a Differential Privacy (DP) baseline and compare it against the proposed encryption framework (see Table IV). As illustrated in Fig. 11, when test data are included in training, the average error (MAE) under DP is 25.854717. When test data are excluded, as shown in Fig. 12, the error increases to 28.467727. However, the average predictive variance remains almost unchanged in both cases (64.574024 vs. 64.645601), indicating that DP noise affects predictive accuracy more strongly than it affects the uncertainty scale.

In contrast, under the proposed method with authorized access (same-key, $p = q$), preserve the core behavior of Gaussian Process inference. In Fig. 3, the model achieves low errors (e.g., Client 1: 12.118) and low variances (around 62–67 across clients), comparable to the plaintext baseline. In Fig. 4, errors increase only modestly (e.g., Client 1: 13.046), while variances increase substantially (e.g., 270.649 and above 200 across clients), reflecting appropriate uncertainty calibration for unseen inputs. This is a key difference from DP: the proposed method retains strong predictive utility under authorized access and still produces a meaningful variance transition between seen and unseen data.

Under the proposed method with unauthorized access (different-key, $(p \neq q)$), Fig. 5 to Fig. 6 exhibit severe degradation: average errors increase to roughly 28–37 and predictive variances explode to the range of $\sim$ 5000–8500. This behavior is desirable from a security perspective because it demonstrates access control: without the correct key, encrypted representations become misaligned, kernel evaluation fails, and accurate prediction becomes infeasible. DP, by design, does not provide such access control; instead, it injects noise to limit information leakage while still producing usable predictions for any party running inference.

Overall, the experimental evidence demonstrates that the proposed key-based encryption framework achieves a significantly better privacy–utility trade-off than Differential Privacy in this federated GPR setting. Unlike DP, which introduces permanent accuracy degradation while providing no access-control mechanism, the proposed method preserves predictive performance under authorized access and simultaneously enforces strict access control under key mismatch. This dual property—high utility for legitimate users and severe degradation for unauthorized users—cannot be achieved by the DP baseline.

F. Multi-Key Access Control Results Across Clients

As shown in Table V, we evaluate whether the proposed encryption pipeline supports key-based access control. When training and testing are performed using the same key, the predictive performance is identical to the plain baseline, confirming that encryption does not degrade utility for authorized users. In contrast, when testing is conducted with a different key or when encrypted training is followed by plain testing, both the prediction error and predictive variance increase substantially. This demonstrates that inference is effectively key-gated: correct keys preserve model usability, whereas incorrect or missing keys result in unreliable outputs, thereby enabling practical multi-key access control.

TABLE V. Multi-Key Access Control Results Across Clients

User (Key)	Train/Test Mode	Avg Error	Avg Variance
User A (Key 1) – *Client 1*	ENC / ENC (in-train, $p = q$)	12.118	62.563
	ENC / ENC (not-in-train, $p = q$)	13.046	270.649
	ENC / ENC (in-train, $p \neq q$)	37.223	6752.542
	ENC / ENC (not-in-train, $p \neq q$)	36.452	8596.941
	ENC / PLAIN (in-train)	31.665	4851.219
	ENC / PLAIN (not-in-train)	31.686	6184.626
	PLAIN / PLAIN (in-train)	12.118	62.563
	PLAIN / PLAIN (not-in-train)	13.046	270.649
User B (Key 3)	*Client 1*		
	ENC / ENC (in-train, $p = q$)	12.118	62.563
	ENC / ENC (not-in-train, $p = q$)	13.046	270.649
	ENC / ENC (in-train, $p \neq q$)	37.223	6752.542
	ENC / ENC (not-in-train, $p \neq q$)	36.452	8596.941
	ENC / PLAIN (in-train)	31.665	4851.219
	ENC / PLAIN (not-in-train)	31.686	6184.626
	PLAIN / PLAIN (in-train)	12.118	62.563
	PLAIN / PLAIN (not-in-train)	13.046	270.649
	Client 2		
	ENC / ENC (in-train, $p = q$)	12.739	64.633
	ENC / ENC (not-in-train, $p = q$)	13.533	277.650
	ENC / ENC (in-train, $p \neq q$)	28.912	6938.149
	ENC / ENC (not-in-train, $p \neq q$)	28.253	8474.300
	ENC / PLAIN (in-train)	27.153	4067.699
	ENC / PLAIN (not-in-train)	26.074	5009.210
	PLAIN / PLAIN (in-train)	12.739	64.633
	PLAIN / PLAIN (not-in-train)	13.533	277.650
	Client 3		
	ENC / ENC (in-train, $p = q$)	12.144	67.212
	ENC / ENC (not-in-train, $p = q$)	12.889	342.483
	ENC / ENC (in-train, $p \neq q$)	34.213	6420.077
	ENC / ENC (not-in-train, $p \neq q$)	34.104	8318.913
	ENC / PLAIN (in-train)	26.935	6188.178
	ENC / PLAIN (not-in-train)	26.351	7849.094
	PLAIN / PLAIN (in-train)	12.144	67.212
	PLAIN / PLAIN (not-in-train)	12.889	342.483

This constant pattern suggests that key accuracy, rather than client-specific data features, is the primary driver of the observed impacts. Consequently, in the federated setup, access control is applied in a key-dependent but client-agnostic manner.

G. Implications for Membership Inference and Robustness

Predictive variance offers a helpful viewpoint for analyzing the results' security implications. Predictive variance is consistently smaller for training samples than for unseen data in both the authorized encrypted configurations and the plaintext baseline, creating a variance gap that may be used for membership inference. However, prediction variance becomes consistently substantial across all inputs under unauthorized access. Thus, variance-based membership inference signals are weakened by successfully suppressing the relative variance difference between training and non-training data. Crucially, this effect does not depend on noise injection or other cryptographic techniques; rather, it naturally results from key-dependent feature-space misalignment. Predictive utility is generally maintained for authorized users at the same time

VI. Discussion and Limitations

The results show that the proposed framework preserves predictive performance under consistent key usage, while mismatched keys lead to large prediction errors and increased predictive variance. This confirms that orthogonal transformation can effectively enforce inference-time access control through representation misalignment.

From a privacy perspective, the uniform increase in predictive variance under key mismatch reduces distinguishable patterns between training and non-training samples, thereby mitigating variance-based membership inference attacks. Compared to existing approaches, differential privacy introduces accuracy loss due to noise, while homomorphic encryption incurs high computational overhead. In contrast, the proposed method maintains accuracy for authorized users without relying on noise or heavy cryptographic operations.

However, several limitations remain. Although evaluated on a real-world dataset, the analysis is limited to a specific domain, and further validation on diverse datasets is required. In addition, the framework focuses on variance-based attacks and does not consider other threats such as model inversion. Finally, the method relies on secure key management, which is not addressed in this work.

VII. Conclusion and Future Work

This research studied inference-time privacy and robustness in federated GPR through predictive uncertainty. We proposed a federated framework based on key-dependent orthogonal feature transformations that enforce multi-key access control at inference time. Experiments show that authorized users retain accurate predictions with well-calibrated uncertainty, while mismatched keys lead to large prediction errors and extreme predictive variance.

In our framework, predictive variance plays a dual role: it functions both as an indicator for key-based access control and as a safeguard against membership inference. When queries originate from authorized users, the model's uncertainty remains properly calibrated. In contrast, unauthorized queries result in uniformly high predictive variance, eliminating the usual variance gap between training and non-training samples. Importantly, this behavior arises without adding artificial noise or relying on heavy cryptographic techniques. Overall, the proposed method achieves a practical balance among model utility, trustworthy uncertainty estimation, and privacy-preserving multi-key authorization in FL environments.

For future work, we plan to evaluate the framework on a broader range of datasets to further assess its generalization properties. We also intend to investigate alternative orthogonal transformation strategies and compare them systematically to analyze their effects on predictive accuracy, uncertainty calibration, robustness, and multi-key access behavior relative to existing approaches.

References

[1] "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data (General Data Protection Regulation)," *Official Journal of the European Union*, 2016. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT [Accessed: Feb. 21, 2026].

[2] H. B. McMahan, E. Moore, D. Ramage, and B. Agüera y Arcas, "Federated Learning of Deep Networks Using Model Averaging," *CoRR*, vol. abs/1602.05629, 2016.

[3] B. Liu, N. Lv, Y. Guo, and Y. Li, "Recent Advances on Federated Learning: A Systematic Survey," *Neurocomputing*, vol. 597, Art. no. 128019, 2024.

[4] J. Chen, H. Yan, Z. Liu, M. Zhang, H. Xiong, and S. Yu, "When Federated Learning Meets Privacy-Preserving Computation," *ACM Computing Surveys*, vol. 56, no. 12, pp. 1–36, 2024.

[5] N. A. Jalali and H. Chen, "Federated Learning Security and Privacy-Preserving Algorithm and Experiments Research Under Internet of Things Critical Infrastructure," *Tsinghua Science and Technology*, vol. 29, no. 2, pp. 400–414, 2023.

[6] A. M. Elbir, B. Soner, S. Çöleri, D. Gündüz, and M. Bennis, "Federated Learning in Vehicular Networks," in *Proc. IEEE Mediterranean Conf. Communications and Networking (MeditCom)*, 2022, pp. 72–77.

[7] A. ElZemity and B. Arief, "Privacy Threats and Countermeasures in Federated Learning for Internet of Things: A Systematic Review," in *Proc. IEEE Int. Conf. Internet of Things (iThings)*, 2024, pp. 331–338.

[8] J. Fu, Y. Hong, X. Ling, L. Wang, X. Ran, Z. Sun, and Y. Cao, "Differentially Private Federated Learning: A Systematic Review," *ACM Computing Surveys*, 2024.

[9] C. Chronis, I. Varlamis, Y. Himeur, A. N. Sayed, T. M. Al-Hasan, A. Nhlabatsi, and G. Dimitrakopoulos, "A Survey on the Use of Federated Learning in Privacy-Preserving Recommender Systems," *IEEE Open Journal of the Computer Society*, vol. 5, pp. 227–247, 2024.

[10] Q. Yang, A. Huang, L. Fan, C. S. Chan, J. H. Lim, K. W. Ng, and B. Li, "Federated Learning with Privacy-Preserving and Model IP-Right Protection," *Machine Intelligence Research*, vol. 20, no. 1, pp. 19–37, 2023.

[11] S. Zhou and G. Y. Li, "Communication-Efficient ADMM-Based Federated Learning," arXiv:2110.15318, 2021.

[12] H. Ye, L. Liang, and G. Y. Li, "Decentralized Federated Learning with Unreliable Communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 487–500, 2022.

[13] R. Shokri and V. Shmatikov, "Privacy-Preserving Deep Learning," in *Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS)*, 2015, pp. 1310–1321.

[14] E. Tabassi, K. J. Burns, M. Hadjimichael, A. D. Molina-Markham, and J. T. Sexton, "A Taxonomy and Terminology of Adversarial Machine Learning," *NIST Interagency/Internal Report (NISTIR)*, pp. 1–29, 2019.

[15] M. Veale, R. Binns, and L. Edwards, "Algorithms That Remember: Model Inversion Attacks and Data Protection Law," *Philosophical Transactions of the Royal Society A*, vol. 376, no. 2133, p. 20180083, 2018.

[16] M. R. Islam, J. F. Akhi, and T. Nakachi, "Defending Against Gaussian Process Membership Inference Attack," in *Proc. 9th Int. Conf. Cryptography, Security and Privacy (CSP)*, 2025.

[17] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A Review of Applications in Federated Learning," *Computers & Industrial Engineering*, vol. 149, Art. no. 106854, 2020.

[18] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282.

[19] W. Du, Y. S. Han, and S. Chen, "Privacy-Preserving Multivariate Statistical Analysis: Linear Regression and Classification," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2004.

[20] M. R. Islam and J. F. Akhi, "A Privacy-Preserving Gaussian Process Regression Framework Against Membership Inference Attacks Using Random Unitary Transformation," *International Journal of Advanced Computer Science & Applications*, vol. 16, no. 8, 2025.

[21] T. Nakachi, Y. Bandoh, and H. Kiya, "Secure Overcomplete Dictionary Learning for Sparse Representation," *IEICE Transactions on Information and Systems*, vol. 103, no. 1, pp. 50–58, 2020.

[22] T. Nakachi and H. Kiya, "Secure OMP Computation Maintaining Sparse Representations and Its Application to EtC Systems," *IEICE Transactions on Information and Systems*, vol. 103, no. 9, pp. 1988–1997, 2020.

[23] Y. Wang and T. Nakachi, "A Privacy-Preserving Learning Framework for Face Recognition in Edge and Cloud Networks," *IEEE Access*, vol. 8, pp. 136056–136070, 2020.

[24] Y. Bandoh, T. Nakachi, and H. Kiya, "Distributed Secure Sparse Modeling Based on Random Unitary Transform," *IEEE Access*, vol. 8, pp. 211762–211772, 2020.

[25] J. F. Akhi, M. R. Islam, and T. Nakachi, "Selective Feature Encryption for Secure Gaussian Process Regression," in *Proc. 13th Int. Conf. Information Management and Engineering (ICIME)*, 2025.

[26] National Center for Health Statistics, "National Health and Nutrition Examination Survey (NHANES), 2021–2023 Data Documentation, Codebook, and Frequencies," Centers for Disease Control and Prevention, 2023. [Online]. Available: https://wwwn.cdc.gov/nchs/nhanes/continuousnhanes/default.aspx?Cycle=2021-2023 [Accessed: Feb. 21, 2026].