cryptography

MDPI

Article

Privacy-Preserving k-Nearest Neighbor Classification over Malicious Participants in Outsourced Cloud Environments

Xian Guo *, Ye Li, Yongbo Jiang, Jing Wang and Junli Fang

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China; ly08122021@163.com (Y.L.); jiangyb@lut.edu.cn (Y.J.); wangjing@lut.edu.cn (J.W.); fangjl@lut.edu.cn (J.F.)
* Correspondence: iamxg@163.com

Abstract: In recent years, many companies have chosen to outsource data and other data computation tasks to cloud service providers to reduce costs and increase efficiency. However, there are risks of security and privacy breaches when users outsource data to a cloud environment. Many researchers have proposed schemes based on cryptographic primitives to address these risks under the assumption that the cloud is a semi-honest participant and query users are honest participants. However, in a real-world environment, users' data privacy and security may be threatened by the presence of malicious participants. Therefore, a novel scheme based on secure multi-party computation is proposed when attackers gain control over both the cloud and a query user in the paper. We prove that our solution can satisfy our goals of security and privacy protection. In addition, our experimental results based on simulated data show feasibility and reliability.

Keywords: privacy-preserving; security; encryption; cloud computing

Citation: Guo, X.; Li, Y.; Jiang, Y.; Wang, J.; Fang, J. Privacy-Preserving k-Nearest Neighbor Classification over Malicious Participants in Outsourced Cloud Environments. *Cryptography* **2023**, *7*, 59. https://doi.org/10.3390/cryptography7040059

Academic Editors: Hanlin Zhang, Zengpeng Li and Dou An

Received: 9 October 2023
Revised: 9 November 2023
Accepted: 15 November 2023
Published: 17 November 2023

1. Introduction

Cloud computing, as a progressive paradigm of information technology, offers the capability of on-demand delivery of diverse computing services such as storage, processing power, and application platforms. On the one hand, many enterprises and even governments have been attracted to using cloud computing to address issues in their business due to its ubiquity, convenience, and on-demand network access [1,2]. On the other hand, security has always been a major issue hindering the widespread adoption of cloud computing technologies [3,4]. The inherent nature of cloud computing causes users to lose control over cloud servers, which may raise concerns and doubts regarding security and privacy risks. Moreover, cloud computing service providers fail to accurately report security vulnerabilities, further exacerbating the situation [5,6]. For instance, Accenture, one of the biggest consulting and management firms in the world, experienced a data breach due to misconfigured AWS storage buckets. It has been reported that LinkedIn, a large enterprise social networking site, suffered a loss of 167 million account credentials in a data leak [7]. The k-nearest neighbor (KNN) query, serving as a fundamental module for data set querying and everyday data mining tasks, has found extensive application in various scenarios such as multi-keyword ranked search [8], network intrusion detection [9], recommended systems [10], etc. Generally, privacy-preserving techniques for the secure outsourcing of KNN classification mainly employ fast multi-party computation and homomorphic encryption. To enable privacy-preserving calculation in secure multi-party computation, non-colluding parties use shared execution computation based on secret values, ensuring that the values or computation results remain inaccessible unless there is collusion. However, existing schemes mostly assume the participating entities to be semi-trusted or trusted and non-colluding. Nonetheless, malicious participants live in the real world to disrupt or steal the model. Therefore, further research is needed to enhance the security of outsourcing schemes and reduce computational overhead under the assumption of malicious adversaries.

In this paper, we aim to efficiently perform privacy-preserving KNN classification over malicious participants in outsourced cloud environments. In the proposed scheme, our contributions are as follows:

1. In this study, we propose a solution based on somewhat homomorphic encryption (SHE) [11,12] to address security and privacy concerns when a user outsources sensitive data to a cloud. This solution, that the outsourced data are encrypted and stored on the cloud, can eliminate potential security and privacy risks. This outsourcing stage's primary goal is to ensure the confidentiality and integrity of sensitive information in cloud environments.
2. Inspired by [13], a secure computation protocol based on garbled circuits is proposed to address the security and privacy issues in data interaction between two clouds. In this protocol, each cloud can only obtain its garbled data and cannot access the raw data from the other cloud, thereby ensuring data confidentiality.
3. The subordinate phrase of maintaining the anonymity of the QU's identity, daring the authentication process, does not appear to modify the subject of a proposed solution. A proposed solution introduces a Fujisaki–Okamoto commitment (FO)-based [14,15] lightweight anonymous two-way authentication protocol between a cloud and a querying user. The authentication process in our novel protocol ensures the anonymity of a query user by using the commitment.
4. Finally, the security model, security definitions, and security requirements in a malicious dyadic cloud environment are given. And we prove that our solution can satisfy our security and privacy protection goals. In addition, our experimental results based on simulated data show feasibility and reliability.

The structure of the rest of the study is as follows: In Section 2, we provide an overview of the relevant paper. Section 3 presents the foundational knowledge and critical notations necessary for understanding the proposed scheme. In Section 4, we introduce our proposed solution's system model, threat model, and design goals. In Section 5, our novel scheme is proposed. Subsequently, in Section 6, we prove the security of the proposed scheme. In Section 7, we experimentally evaluate the performance of the proposed scheme. Finally, in Section 8, we conclude the paper along with future work.

2. Related Work

KNN has been extensively studied as one of the fundamental operations in data mining and machine learning. In recent years, significant research has been conducted on privacy-preserving KNN queries and classification to safeguard data security and query privacy.

The paper [16] proposed an asymmetric inner product-preserving encryption (ASPE) algorithm to encrypt the original data points in the data set, which uses a reversible random matrix as the encryption and decryption key for the data points. Then, the authors proposed a series of different schemes to enhance the security of the ASPE scheme. Although this scheme partially solves the secure outsourcing problem of KNN classification, it assumes that the querying user is entirely trustworthy and shares the private decryption key of the data owner. Under this assumption, attackers only need to collude with any querying user to decrypt the ciphertext data set, which poses significant security risks.

The paper [17–19] considered the untrustworthy behavior of querying users. It proposed using the Paillier encryption algorithm to encrypt the querying data points, thereby ensuring that querying users could not obtain the data owner's private decryption key. To further enhance the scheme's security, the paper [20] designed a secure KNN query outsourcing scheme using a additive homomorphic cryptosystem, achieving data privacy, query privacy, and result privacy. However, the high cost of querying users limits its practical application in the real world.

Recently, the paper [21] proposed a more efficient location-based KNN cloud query scheme, which uses an improved Paillier homomorphic encryption technique to resist rainbow attacks. None of these studies considered the integrity of the results. Other research has

focused on designing secure protocol building blocks such as those utilizing the ElGamal encryption algorithm in the paper [22] or protocols for secure frequency and computation of majority class in the paper [23]. Subsequent schemes [24–26] have sought to enhance the security and efficiency of outsourced KNNs but have faced computational overhead challenges associated with query data point encryption, reducing their practicality.

To improve the practicality of the scheme, the paper [27] put forward a plan that attains database security, data owner key secrecy, query privacy, and data access pattern hiding. In this scheme, data owners do not participate in the online process, the computational burden on querying users is minimal, and the cloud server can efficiently perform encrypted KNN classification. The paper [28] proposed a KNN query protocol that ensures both high accuracy and efficiency while maintaining the privacy of spatial data. By applying cryptographic transformations based on Moore curves, the protocol achieves efficient KNN queries and protects sensitive information without compromising accuracy. The paper [29] proposed an efficient KNN query and classification scheme based on the K-dimensional tree for outsourced data privacy protection. In this scheme, the original data are encrypted using order-preserving encryption (OPE) and the Paillier cryptosystem, and the data are indexed using kd-tree technology to improve query speed. The paper [30] proposed a KNN set similarity search scheme in cloud computing environments that is efficient, secure, and verifiable. The scheme utilizes one-way hash functions and homomorphic encryption techniques to ensure the correctness and integrity of query results without revealing the privacy of the data, and it has good scalability. The paper [31] proposed an efficient and secure cloud computing SEKNN query scheme based on obfuscation circuits, secret sharing, and Yao protocols to build a more secure and efficient sorting algorithm. It aims to solve the problems of inefficiency and suspicious security in existing articles. This scheme can achieve fast and secure KNN queries in a semi-honest model in cloud computing.

In the subsequent research, numerous secure approximate KNN query schemes have been proposed by researchers. Another approach to preserving privacy while searching encrypted data sets outsourced to the cloud is outlined in [32]. It enables the cloud to recognize the KNN data points within the encrypted data set that are near the encrypted user query. The querying user is then provided with the returned search results. This scheme claims that the data owner always retains the keys and does not need to be shared with others. It aims to overcome the limitations of earlier cutting-edge works, such as the need for key distribution by the data owner and the provision of some storage at the querying user end using non-collusive cloud servers. Nevertheless, this method compromises the security level by inadvertently exposing the access pattern of the data to the cloud server despite ensuring the privacy of the outsourced data set and queries. Based on their research, it is questionable whether the exposure of the data access pattern is a concern when the data are encrypted. Additionally, the system employs an AES algorithm to encrypt the original data. According to cryptographic transformation, this method reduces communication overhead. It thus can effectively provide the KNN of a query while preserving the privacy of location information and spatial data. However, the encryption used by this scheme is AES, which is semantically insecure and raises additional privacy issues. The ASPE technique is utilized in [33] to design an effective scheme for secure similarity search. The system effectively compares the similarity between two points using the characteristics of ASPE, constructs an index based on B+ trees, and achieves an efficient similarity search. However, this scheme does not support result verification and cannot guarantee the integrity of results. The paper [34] proposes a fully non-interactive KNN algorithm based on fully homomorphic encryption, with complexity quadratic in terms of the database size. However, it assumes that most voting is conducted in plaintext, a significant security vulnerability. The paper [35] introduced a new method of addressing privacy and authentication security issues in KNN queries. This method protects data privacy and identity verification by applying cryptography and encryption techniques. However, it is essential to implement extra security measures to protect keys and authentication credentials from potential malicious attacks. HE-V is introduced in [36]

for conducting KNN on encrypted data, incorporating majority voting for assigning class labels. The proposed solution encompasses every step of the KNN algorithm and can handle encrypted data comprehensively, guaranteeing no information leakage throughout the procedure.

In the research work on multi-query user search, a protocol for secure computation using query users' multiple keys for KNN queries is proposed in [37]. In this scenario, the data owner and each query user possess individual keys, eliminating the need for key sharing between them. While data privacy and query attributes are maintained, and the data owner's direct involvement is avoided, the access pattern remains unprotected. In the context of multi-query user settings, more work must be performed to guarantee data security and result integrity effectively. Based on this problem, a secure and verifiable KNN query for multi-query users is proposed in [38]. MSVKNN strives to achieve precise outcomes while safeguarding the confidentiality of data, questions, results, and access patterns and guaranteeing the accuracy and entirety of results in multi-query user scenarios. A proposal has been made for implementing a verifiable, secure index to facilitate private search and result verification for users with multiple queries.

In conclusion, all the research is based on the operation of the cloud in a semi-honest model, while the querying users are assumed to be in an honest model. However, in the real world, we still encounter malicious actors and the possibility of semi-honest users during querying. Therefore, this paper aims to investigate how to enhance the security of outsourcing schemes under the assumption of the cloud being maliciously exploited and querying users being dishonest. Taking into consideration real-world scenarios will make the research more comprehensive and practical.

3. Preliminary

In this section, we present some essential preliminary concepts for our scheme.

3.1. Somewhat Homomorphic Encryption

Somewhat homomorphic encryption (SHE) is a family of algorithms that can perform both additive and multiplicative homomorphic encryption with a limited number of operations—if operations are allowed for an arbitrary time. The limiting factor is the divergence of noise introduced into the ciphertext, primarily by multiplication. The algorithm proposed in Reference [11] is employed as the implementation scheme for SHE in this paper. The plaintext space of our scheme is the polynomial ring $\mathbb{R}_p = \mathbb{Z}_p[X]/\Phi_u(x)$, where p is a large prime and $\Phi_u(x)$ is a u-order cyclotomic polynomial. The SHE scheme comprises the key generation algorithm, the encryption algorithm, and the decryption algorithm. The key generation algorithm is a probabilistic algorithm that inputs the security parameters u, p, and L and produces a key pair (PK, SK) as output. Note that u and p determine the plaintext space, and L determines the depth to which SHE can support homomorphic multiplication. The ENC is a probabilistic algorithm that takes PK and plaintext m as inputs and outputs in the ciphertext $E_{PK}(m)$. The DEC algorithm is deterministic, with the input being the ciphertext $E_{PK}(m)$ and the private key SK and the output being the plaintext m. SHE possesses the following homomorphic properties:

1. Homomorphic addition: Given two ciphertexts $E_{PK}(x)$ and $E_{PK}(y)$, there exists an operation $\oplus$ such that $E_{PK}(x+y) = E_{PK}(x) \oplus E_{PK}(y)$. Given a ciphertext $E_{PK}(x)$ and an open constant α, $E_{PK}(x+\alpha) = E_{PK} \oplus \alpha$.
2. Homomorphic multiplication: Given two ciphertexts $E_{PK}(x)$ and $E_{PK}(y)$, there exists an operation $\otimes$ such that $E_{PK}(x \times y) = E_{PK}(x) \otimes E_{PK}(y)$. Given a ciphertext $E_{PK}(x)$ and an open constant α, $E_{PK}(x \times \alpha) = E_{PK} \otimes \alpha$.

The SHE scheme supports single-instruction multiple data (SIMD) [12], which can encode several messages into a single ciphertext, and the operations on the ciphertext can be applied to several different messages simultaneously. Assume that PACK denotes the pack operation that can pack l different messages $x = (x_1, x_2, \ldots, x_l) \in \mathbb{Z}_p^l$ into plaintext space $\mathbb{R}_p$ element $x^{'}$; unpack denotes the unpack operation that restores $x^{'}$ to

$x = (x_1, x_2, \cdots, x_l)$. Given an arbitrary vector $(x, y, \alpha) \in \mathbb{Z}_p^l$, computer $x^{'} = PACK(x)$, $y^{'} = PACK(x)$, $\alpha^{'} = PACK(\alpha)$, we encrypt $x^{'}, y^{'}$ to obtain $E_{PK}(x^{'})$, $E_{PK}(y^{'})$, and the following equation exists.

$$\begin{aligned} &unpack(DEC(E_{PK}(x^{'}) \oplus E_{PK}(y^{'}))) = (x_1 + y_1, x_2 + y_2, ..., x_l + y_l) \\ &unpack(DEC(E_{PK}(x^{'}) \oplus \alpha)) = (x_1 + \alpha_1, x_2 + \alpha_2, \cdots, x_l + \alpha_l) \end{aligned} \tag{1}$$

$$\begin{aligned} &unpack(DEC(E_{PK}(x^{'}) \otimes E_{PK}(y^{'}))) = (x_1 \times y_1, x_2 \times y_2, \cdots, x_l \times y_l) \\ &unpack(DEC(E_{PK}(x^{'}) \otimes \alpha)) = (x_1 \times \alpha_1, x_2 \times \alpha_2, \cdots, x_l \times \alpha_l) \end{aligned} \tag{2}$$

3.2. Elliptic Curves

An elliptic curve (ECC) defined over a finite field $\mathbb{Z}_p$ is denoted by $E = p, a, b, G, n$. All points on the curve satisfy the equation

$$\begin{aligned} &y^3 = x^3 + bx + a(\mathrm{mod}\, p) \\ &\forall a, b \in \mathbb{Z}_p, 4b^3 + 27a^2 \neq 0(\mathrm{mod}\, p) \end{aligned} \tag{3}$$

where G is a point (of order n) on the curve E, called the generating point.

3.3. Fujisaki–Okamoto Commitment Agreement

Let s be a safe number and h a large composite number. Neither Alice nor Bob knows the factorization of h, $m \in \mathbb{Z}_{n^2}^*, n \in (m)$. Alice does not know the orders of $\log_g h$ and $\log_h g$, with g and h being prime numbers greater than 160 bits, making it infeasible to compute the discrete logarithm in the cyclic group they generate. Alice picks $r \in (2^s h + 1, \cdots, 2^s h - 1)$ at random, computes $F(x, r) = m^x n^r \bmod h$, and sends $F(x, r)$ to x with respect to m and n. This commitment scheme is statistically safe. On the one hand, Alice cannot find the discrete logarithm that makes $x_1 \neq x_2, F(x_1, r_1) = F(x_2, r_2)$ unless it can decompose n or can pick up the discrete logarithm; on the other hand, $F(x, r)$ does not statistically reveal any information to Bob.

It is said that the commitment value of x and $F(x, r)$ is a commitment to x since only Alice knows the commitment value of $F(x, r) = m^x n^r \bmod h$ and the random number (x, r).

3.4. Non-Contact Commitment

The concept of non-contact commitment was first introduced by the authors of [39], and later the authors of [40] designed a mechanism for bit commitment based on it. In this article, the focus is on the verification of each participant's information in conjunction with a non-contact commitment mechanism.

This mechanism has the following characteristics:

Correctness: For all CRS, if $if(C, \delta) \leftarrow COM_{CRS}(x)$, then $CHK_{CRS}(C, \delta) = x$.

Binding: For all polynomial time attackers A, it can output $(C, \delta, \delta^{'})$ with a negligible probability of $A(CRS)$, so that $CHK_{CRS}(C, \delta) \neq CHK_{CRS}(C, \delta^{'})$ and $\perp \notin CHK_{CRS}(C, \delta) \neq CHK_{CRS}(C, \delta^{'})$. It can be ignored with probability $A(CRS)$ that the output $(C, \delta, \delta^{'})$ gives $CHK_{CRS}(C, \delta) \neq CHK_{CRS}(C, \delta^{'})$ as well as $\perp \notin CHK_{CRS}(C, \delta) \neq CHK_{CRS}(C, \delta^{'})$.

Hiddenness: For all polynomial time attackers A, all of CRS and $(x, x^{'}) \in 0, 1_n$ are ignored as below.

$$\| Pr_{(C,\delta) \leftarrow COM_{CRS}(x)}[A(C) = 1] - Pr_{(C,\delta^{'}) \leftarrow COM_{CRS}(x^{'})}[A(C) = 1] | \tag{4}$$

3.5. Garbled Circuits

For any function f, garbled circuits allow two participants to securely compute the function value $f(x, y)$ without revealing their respective holdings x and y. The core idea of a garbled circuit is that one party (the circuit generator) first encrypts the Boolean circuit

corresponding to function f and its input x and sends the encrypted Boolean circuit and input to the other party (the circuit calculator). The circuit computation side interacts with the circuit generation side to obtain the encrypted value of its input y. Then, it performs operations on the encrypted Boolean circuit in combination with the encrypted value of the received x. The result is that both parties can compute the function value $f(x, y)$ without obtaining valid information from each other's inputs.

4. System Model, Threat Model, and Design Goals

In this section, we present the notations, system model, threat model, and design goals of the scheme.

4.1. Notations

We give the notations used in this paper as shown in Table 1.

Table 1. Notations.

Notation	Description
$\oplus$	homomorphic addition
$\otimes$	homomorphic multiplication
D	data set of n labeled samples
d	each sample dimension in the data
t, l, s_1, s_2	the secret parameters
m_1, m_2, n_1, n_2	random numbers selected in the cyclic group
q	a private plain query point for the query user
r_1, r_2, x	user's secret parameters
q_1, q_2, ω	random parameters
N_C	a random number
Com_1, Com_2	the user's commitment computed
PK_{QU}	the QU's public key
SK_{QU}	the QU's private key
$q^{'}$	q of the encrypted query points
$c_q^{'}$	encrypted KNN classification label set
c_q	KNN categorical tag set
$\mathbb{R}_p = \mathbb{Z}_p[X]/\Phi_u(x)$	polynomial ring
K_d	session key

4.2. System Model

In this paper, our privacy-preserving KNN classification system involves two non-colluding clouds, data owners, and query users, as shown in Figure 1.

1. Dyadic cloud: In our system, two independent clouds are denoted as C_A and C_B, respectively. They both maintain the data provided by the data owners. Through a series of secure protocols, they can answer multiple queries in a way that preserves privacy.
2. Data owner (DO): Our solution requires that the DO generates and encrypts its data with SHE and then uploads the encrypted data to a cloud.
3. Query user (QU): An anonymous two-way authentication protocol based on Fujisaki–Okamoto commitment is adopted between a cloud and the QU. Next, our solution requires that a QU submits its encrypted sample to a cloud when it needs to query some information. A cloud will respond to the QU's query when it receives a QU's request—the QU recovery query results.

It is important to note that these two non-colluding clouds are practical in a commercial environment. For instance, a C_A may belong to Amazon, while a C_B could be Microsoft's Azure. Once colluding with each other is discovered by others, customers can lose trust in either of them, leading to a significant loss in market share for both companies. Various privacy-preserving machine learning schemes have adopted this non-collusive dual-cloud model, as it can effectively minimize QUs' computational and communication costs [41].

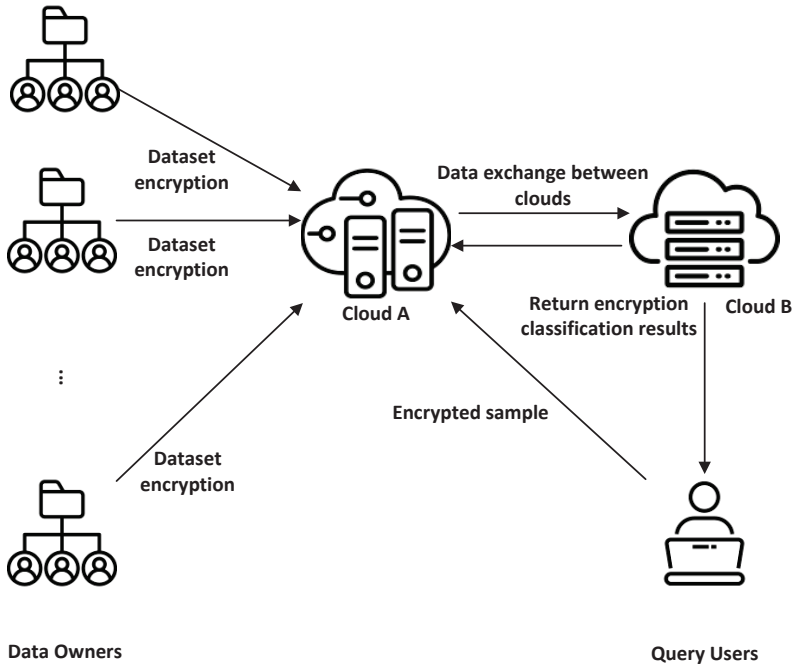

Figure 1. System model.

4.3. Threat Model

In our scheme, we assumed that the C_A is a semi-honest participant, the C_B is a malicious participant, and some QUs are semi-honest participants.

1. The malicious C_B may illegally collect and infer the true identity of a QU, and the C_B can reconstruct sensitive information based on storage location information. In an outsourcing system, the C_B can arbitrarily deviate from the specified computation task. It not only tries to capture users' private information but also may return a deliberately forged result to fool the QU.
2. The semi-honest QU that an attacker compromised may access the system by impersonating a legitimate QU, and it also performs a replay attack or observes messages between a cloud and a legitimate QU to capture another QU's privacy.

4.4. Design Goals

This article proposes a scheme to enhance security in the query and outsourcing process. Based on associated security protocols such as garbled circuits, homomorphic encryption, and anonymous authentication, the scheme eliminates threats in the query and outsourcing process. Our new plan fulfills the following security objectives.

1. Privacy of data outsource: A DO's plaintext data and private key are uniquely known only to itself and will not be disclosed to other participating parties.
2. Privacy of query data: The data access pattern in the ciphertext KNN classification process ensures that the data points in the ciphertext data set corresponding to the KNN classification labels are not disclosed to the cloud. Additionally, the authentication process between a QU and the cloud ensures the confidentiality of the real identity. While adversaries can obtain the pseudonym of the QU, they cannot deduce any relevant information about the real identity of the QU, thereby preventing potential privacy leakage risks.
3. Privacy of interaction between clouds: In the protocol of secure mean calculation, data privacy is ensured as each cloud can only access its own obfuscated data, thereby preventing access to the original data of another cloud.

5. Our Scheme

In this section, we propose a novel KNN classification scheme with enhanced privacy protection, which includes four parts: data outsourcing, query data, data interaction between clouds, and query result recovery.

5.1. Plaintext Encoding

The SHE technique is utilized to encrypt outsourced data. A problem that needs to be addressed when we use SHE for classification computation is that the plaintext space of the SHE algorithm is a polynomial ring $\mathbb{R}_p = \mathbb{Z}_p[X]/\Phi_u(x)$, while the data in the outsourced data typically include floating-point numbers and negative numbers. Therefore, our approach converts the data to $\mathbb{Z}_p$ and then utilizes the PACK operation in SHE to encode l different data $a = (a_1, a_2, \cdots, a_l) \in \mathbb{Z}_p^l$ into the plaintext space $\mathbb{R}_p$. In particular, for a floating-point or integer value x, we select a large integer γ multiplied by x and then round the result to an integer $(p + x) \in \mathbb{Z}_p$. For negative integer x, it is denoted as $(p + x) \in \mathbb{Z}_p$ in this study.

One additional issue to address when using SHE for classification computation is the prevention of result overflow during the calculation process. Our approach combines the Chinese remainder theorem (CRT) to tackle this problem. Specifically, we select multiple prime numbers $p_1, p_2, \cdots, p_h$ as the modulus of the plaintext space for the SHE and generate h corresponding public–private key pairs $(PK_i, SK_i, i \in [1, h]$. For a large integer $x > p_i$, it is encoded into the plaintext space $\mathbb{R}_{p_i}$ and encrypted using PK_i to obtain $E_{pk_i}(x)$. In our notation, we represent the ciphertext of x under h different keys as $E_{pk}(x) = (E_{pk_1}(x), E_{pk_2}(x), \cdots, E_{pk_h}(x))$.

5.2. Security of Data Outsourcing

In this stage, each DO uses SHE technology to encrypt their respective data and encrypts them to upload them to the C_A.

The implementation steps are as follows:

- Step 1: Select an element from a discrete Gaussian distribution $s \leftarrow \chi, s \in R$.
- Step 2: Select an element $p_i, e, p_i \in R_p$ from R_p.
- Step 3: Set the public key to $PK = (p_0, p_1)$, where $p_0 = -(p_1 s + te)$, and the private key is $SK = s$.
- Step 4: A DO utilizes the expansion factor γ to expand and round each of the sample data and transforms each sample's class label C_i into a 0-1 vector c_i. If the condition $C_i = l_t$ is satisfied, the t-th position in c_i is set to 1, while the other bits are set to 0.
- Step 5: The DO divides data value D into s blocks, each containing l samples. We assume the total number of samples n is a multiple of l. For each data block i, the DO bundles l data records with the same attribute j and encrypts them with public key PK to obtain A_{ij}.
- Step 6: The DO encrypts the class labels corresponding to each sample in data block i to obtain C_{it}.
- Step 7: The DO receives and transmits the encrypted data $E_{pk}(D)$ to C_A.

In addition, the data owner generates a random permutation function φ for n data, utilizes it to compute set $\zeta^{'} = \varphi(x)$, and sends the combined set φ, which consists of the function φ and the randomly permuted data, to C_A for subsequent encrypted KNN classification. After the completion of data set outsourcing, the DO remains offline during the ciphertext KNN classification process.

5.3. Query Privacy

To preserve the identity of a QU during the authentication process, we designed an anonymous mutual authentication protocol based on the FOC protocol. The essence of a QU can be accurately verified by verifying the proof that the user holds P, as validated by C_A and C_B. We employ ECC cryptography for mutual authentication during the

authentication procedure. Furthermore, zero-knowledge proof (ZKP) is utilized to conceal the user's identity.

In our scheme, we let t, l, s_1, s_2 denote four security parameters and h denote a large composite number factorization C_A, C_B, which the QU is unaware of. m_1 and m_2 are the two elements with the highest order in $\mathbb{Z}_h$. n_1 and n_2 are elements in the m_1 and m_2 generating groups. The QU is unaware of $\log_{m_1} n_1, \log_{n_1} m_1, \log_{m_2} n_2, \log_{n_2} m_2$. H is a hash function. The QU secretly holds $y \in [0, a]$. Let $Com_1 = Com_{11}(y, r_1)$ and $Com_2 = Com_{12}(y, r_2)$ be two commitments relating to y. In order to demonstrate the QU's knowledge of y, r_1, r_2, the C_A and the C_B can verify the QU's commitment $\{Com_1, Com_2\}$, in which $\{Com_1, Com_2\}$ conceals the same secret y.

- Step 1: The QU randomly selects $\omega \in [1, 2^{l+tb} - 1], \eta_1 \in [1, 2^{l+t+s_1}h - 1]$, $\eta_2 \in [1, 2^{l+t+s_2}h - 1]$, by performing the following computation.

$$W_1 = m_1^{\omega} n_1^{\eta_1} \bmod h \tag{5}$$

$$W_2 = m_2^{\omega} n_2^{\eta_2} \bmod h \tag{6}$$

- Step 2: The QU calculates $c = H(W_1 \parallel W_2)$. Then, the QU computes

$$I = \omega + cx \tag{7}$$

$$I_1 = \eta_1 + cr_1 \tag{8}$$

$$I_2 = \eta_2 + cr_2 \tag{9}$$

  and sends $\{c, I, I_1, I_2\}$, respectively, to C_A and C_B.
- Step 3: The C_A and the C_B check if c is satisfied.

$$c = M(m_1^I n_1^{I_1} Com_1^{-c} \bmod h \parallel m_2^I n_2^{I_2} Com_2^{-c} \bmod h) \tag{10}$$

If the protocol is successfully executed, the C_A and the C_B will believe that Com_1 and Com_2 have concealed the same secret number. If $1/l$ is negligible, in the random oracle model, the protocol achieves statistical zero knowledge.

The implementation steps are as follows:

- Step 1: The C_A and the C_B, respectively, randomly generate N_C and send them to the QU that is requested for signature for anonymous authentication.
- Step 2: After receiving the N_C, the QU randomly selects K_d to serve as a session key. Then, we generate a proof P for the user's identity in the following manner:

$$P = \{c, I, I_1, I_2\} \tag{11}$$

  where $\{c, I, I_1, I_2\}$ is calculated via the following equation.

$$\begin{aligned} c &= H(W_1 \parallel W_2 \parallel N_C) \\ I &= \omega + cy \\ I_1 &= q_1 + cr_1 \\ I_2 &= q_2 + cr_2 \end{aligned} \tag{12}$$

- Step 3: According to the equation's results, P is obtained for the QU.
- Step 4: The QU encrypts $\{K_d, c, I, I_1, I_2\}$ with the public key PK_H and obtains

$$\lambda = Enc_{K_u}(K_d, c, I, I_1, I_2) \tag{13}$$

Then, the QU sends λ to the C_A and the C_B.

- Step 5: The C_A and the C_B first decrypt the result of λ with the private key to evidence the QU and the K_d. Unauthorized cloud entities should not access the secret y stored by I. When the P provided by the QU is accurate, the following equation holds according to the FOC protocol. $W_1^{'} = W_1, W_2^{'} = W_2$, where

$$\begin{aligned} W_1^{'} &= m_1^I n_1^{I_1} F^{-c} \bmod h \\ W_2^{'} &= m_2^I n_2^{I_2} E^{-c} \bmod h \end{aligned} \tag{14}$$

Then, the C_A and the C_B can verify the QU according to the following equation.

$$c = H(m_1^I n_1^{I_1} Com_1^{-c} \bmod h \parallel m_2^I n_2^{I_2} Com_2^{-c} \bmod h \parallel N_C) \tag{15}$$

- Step 6: If the above verification is correct, then the QU is legal. The C_A and the C_B use the K_d to encode the message and transmit it to the QU for encryption.
- Step 7: If the QU can decrypt the messages sent by the C_A and the C_B with the K_d, then the C_A and the C_B are secure, and authentication is concluded.

Query data encryption
We let the query data point of the QU be $q = (q_1, q_2, \cdots, q_d), q_i \in \mathbb{Z}_h$.

- Step 1: The QU initially generates a random number that is positive in value $\varepsilon \in \left\{1, \cdots, 2^{l_2}\right\}$ and converts q into a $d+1$-dimensional vector $\hat{q} = \varepsilon(q, 1)$.
- Step 2: The QU encrypts each element $\hat{q}^{'}$ of vector $\hat{q}$ with the PK_{QU} obtaining the ciphertext query data points and sends them to the C_A for ciphertext KNN classification.

$$q^{'} = (E_{PK}(\hat{q}_1), E_{PK}(\hat{q}_2), \cdots, E_{PK}(\hat{q}_{d+1})) \tag{16}$$

Note: Before performing SHE encryption, data point p_i and query data point q in the data are transformed into d+1-dimensional vectors $\hat{p}_i$ and $\hat{q}$, respectively. The reason for the transformation is to use inner product instead of Euclidean distance $\parallel p_i - q \parallel$ as the basis for determining the similarity between data points in KNN, thereby reducing the computational cost of encrypted KNN classification.

5.4. Security of Data Exchange among Clouds

The bit commitment mechanism in our scheme is mainly based on the universal random string model. We let CRS be a universal random string and (COM_{CRS}, CHK_{CRS}) be a non-contact commitment mechanism for n-bit messages. The COM_{CRS} algorithm selects n-bit message x and random coin toss r as inputs, and as outputs commitment C and its corresponding path δ. Notation $COM_{CRS}(x)$ is shorthand for $COM_{CRS}(x; \beta)$ based on β.

A four-value pair algorithm $G = (Gb, En, De, Ev, ev)$ represents the garbled circuit, as shown in Figure 2.

Among them, Gb represents a random garbled algorithm that can convert the calculation of f into a ternary pair of (F, e, r). Here, F represents a garbled circuit, where e is the encrypted information and r is the decrypted information. En is the encryption algorithm that maps data value x to input $X = En(e, x)$ of the garbled algorithm. De is the decryption algorithm $y = De(r, Y)$ that can restore the output Y of the garbled algorithm to the data value y. Ev is the algorithm for verifying whether the input X of the garbled algorithm and the garbled circuit F meet the output $Y = Ev(F, X)$ of the garbled algorithm. The correctness of the garbled circuit refers to all (F, e, r) and input information X supported by the garbled algorithm $Gb(1^k, f)$, where $De(r, Ev(F, En(e, x))) = f(x)$ and k represent security parameters.

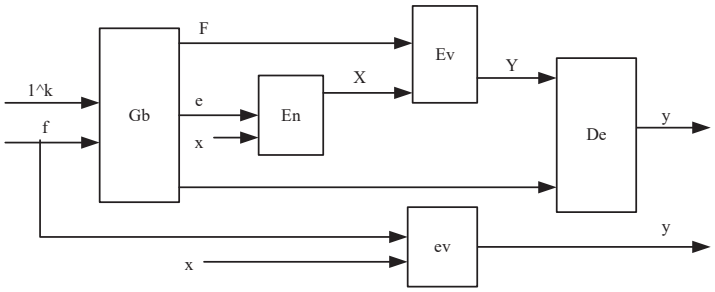

Figure 2. Structure of the garbled circuit.

A secure average calculation protocol based on garbled circuits aims to obtain the final calculation result in the presence of malicious participants (the C_B). This protocol has three participants: the C_{A_1}, the C_{A_2} (the C_{A_1} and the C_{A_2} are the two servers of the C_A which are used to verify whether the C_B is a malicious participant), and the C_B. The C_{A_1} inputs (x_1, t_1), the C_{A_2} inputs (x_2, t_2), and the C_B is mainly responsible for the circuit calculation process, intending to safely calculate $y = x_1 + x_2/t_1 + t_2$.

The implementation steps are as follows:

- Step 1: The C_B randomly selects CRS for the promise and, respectively, sends CRS to the C_{A_1} and the C_{A_2}. The C_B generates random number ξ and shares the secret as $\xi = \xi_1 \oplus \xi_2$, sends ξ_1 to the C_{A_1}, and sends ξ_2 to the C_{A_2}.
- Step 2: The C_{A_1} selects seed $\beta \leftarrow \{0,1\}^k$ for pseudo-random function rpf and then sends β to the C_{A_2}.
- Step 3: The C_{A_1} and the C_{A_2} generate the corresponding circuit $Gb(1^k, f) \rightarrow (F, e, r)$ based on function $f = (x_1 + x_2)\xi$ and randomly select $a_1, a_2 \leftarrow \{0,1\}^{4m}$, for all $\mu \in 4m$ and $v \in \leftarrow \{0,1\}$ to generate the following commitments:

$$\begin{aligned}(C^v_{1,\mu}, \delta^v_{1,\mu}) &\leftarrow COM_{CRS}(e[\mu, a_1[\mu] \oplus v]) \\ (C^v_{2,\mu}, \delta^v_{2,\mu}) &\leftarrow COM_{CRS}(e[\mu, a_2[\mu] \oplus v])\end{aligned} \tag{17}$$

Finally, the C_{A_1} and the C_{A_2} send to the C_B the following message:

$$\begin{aligned}&(a_1[2m+1 \cdots 4m], F, \left\{C^v_{1,\mu}\right\}_{\mu,v}) \\ &(a_2[2m+1 \cdots 4m], F, \left\{C^v_{2,\mu}\right\}_{\mu,v})\end{aligned} \tag{18}$$

- Step 4: If the information sent by the C_{A_1} and the C_{A_2} is different, then the C_B will stop.
- Step 5: The C_{A_1} and the C_{A_2} send uncommitted messages $\delta^{x_1[\mu]\oplus a_1[\mu]}_{1,\mu}$, $\delta^{\xi_1[\mu]\oplus a_1[2m+\mu]}_{1,2m+\mu}$, $\delta^{t_1[\mu]\oplus a_2[\mu]}_{2,\mu}$, $\delta^{\xi_1[\mu]\oplus a_2[2m+\mu]}_{2,2m+\mu}$ and $\delta^{x_2[\mu]\oplus a_1[m+\mu]}_{1,m+\mu}$, $\delta^{\xi_2[\mu]\oplus a_1[3m+\mu]}_{1,3m+\mu}$, $\delta^{t_2[\mu]\oplus a_2[m+\mu]}_{2,m+\mu}$, $\delta^{\xi_2[\mu]\oplus a_2[3m+\mu]}_{2,3m+\mu}$, respectively, to the C_B.
- Step 6: For $\mu \in 4m$, the C_B for any correct $o[\mu]$ are used to calculate:

$$X[\mu] = CHK_{CRS}(C^{o[\mu]}_{1,\mu}, \delta^{o[\mu]}_{1,\mu}) \tag{19}$$

$$X^{'}[\mu] = CHK_{CRS}(C^{o[\mu]}_{2,\mu}, \delta^{o[\mu]}_{2,\mu}). \tag{20}$$

If there is a CHK call, then return $\perp$, and then stop. Similarly, the C_B knows the data $a_1[2m+1 \cdots 4m]$ and $a_2[2m+1 \cdots 4m]$, and the protocol stops if the C_{A_1} or the C_{A_2} cannot unlock ξ_1 and ξ_2 corresponding to the promise $C^{\xi_1[\mu]\oplus a_1[2m+\mu]}_{1,2m+\mu}$, $C^{\xi_2[\mu]\oplus a_1[3m+\mu]}_{1,3m+\mu}$, $C^{\xi_1[\mu]\oplus a_2[2m+\mu]}_{2,2m+\mu}$, $C^{\xi_2[\mu]\oplus a_2[3m+\mu]}_{2,3m+\mu}$. Then, C_B executes $Y \leftarrow Ev(F, X)$, $Y^{'} \leftarrow Ev(F, X^{'})$ and sends Y and $Y^{'}$ to the C_{A_1} and the C_{A_2}.

- Step 7: The C_{A_1} and the C_{A_2} calculate whether $x_1 + x_2/t_1 + t_2 = De(r, Y)/De(r, Y^{'})$ holds.

Ciphertext KNN classification

In this protocol, after receiving the ciphertext query sample from QU, C_A and C_B jointly perform the ciphertext KNN classification task based on the ciphertext database uploaded by DO.

The C_A performs the following steps:

- Step 1: The C_A calculates the ciphertext inner product $dst_i^{'}$ of the ciphertext query sample $q^{'}$ and the ciphertext data point $p_i^{'}$ in the ciphertext data, and it obtains the following set of ciphertext inner products.

$$dst_i^{'} = \left\{ dst_1^{'}, dst_2^{'}, \cdots, dst_n^{'} \right\}$$
$$dst_i^{'} = p_i^{'}(q^{'})^T \tag{21}$$

- Step 2: The C_A arranges the elements in the set of ciphertext inner product $dst_i^{'}$ and the set of ciphertext classification labels V by using the random permutation function Θ, and it obtains the following two sets of random permutations.

$$dst^{'} = \Theta(dst^{'}) = \left\{ dst_1^{'}, dst_2^{'}, \cdots, dst_n^{'} \right\} \tag{22}$$
$$V = \Theta(V) = \{v_1, v_2, \cdots, v_n\} \tag{23}$$

The C_A sends collection dst and V to C_B.

The C_B performs the following steps:

- Step 1: The C_B uses the received ciphertext inner product set $dst^{'} = \left\{ dst_1^{'}, dst_2^{'}, \cdots, dst_n^{'} \right\}$ for KNN search.
- Step 2: The C_B bases on the structure obtained by using the ciphertext inner product $dst^{'}$ for KNN search, and the ciphertext KNN classification label set $C_q^{'} = \{v_1, v_2, \cdots, v_n\}$ is obtained from the ciphertext classification label set V and sent to the QU.

5.5. Query Result Recovery

After receiving the ciphertext KNN classification label $C_q^{'}$ sent by C_B, for each ciphertext label in the set, a QU decrypts it using SHE SK_{QU} to obtain the KNN classification label set C_q.

6. Security Analysis

In this section, the security of our solution will be analyzed.

6.1. Data Outsourcing Security

Theorem 1. *SHE-based encryption is secure in data transfer.*

Proof. The data transferred between the user and the server are encrypted. Entrusting data to an untrusted third-party organization or being subjected to hacker attacks can result in data leakage. The SHE scheme provides strong protection to the data, making it impossible for attackers to recover the original data. All classified computations are performed within the encrypted domain, eliminating the need for decryption. As a result, the scheme offers higher security for remote data storage and leak prevention. □

6.2. Query Privacy Security

Definition 1. *There exists an efficient algorithm that, when given input $|h|$, generates an RSA mod n and an associated element $z \in \mathbb{Z}_p$ ($\mathbb{Z}_p$ is multiplicative group), where $e \in \{0, 1\}$ and integer $z = u^e$ mod h, if satisfied, is infeasible.*

Definition 2. *Finding the integer $l, 0 \leq l \leq h-1$ such that $Q = lP$ is a challenging task, given an elliptic curve Com_1 defined over a finite field Com_{2_q} and a point $P \in Com_1(Com_{2_q})$ of order $l, 0 \leq l \leq h-1$.*

Theorem 2. *If the probability of C_A, C_B, or QUs generating each other's private keys is negligible in our proposal, the attacker will be unable to successfully impersonate a legitimate C_A, C_B, or QU.*

Proof. Under the condition of the robust RSA assumption, as inferred by [4], $Com_1(y, r)$ is considered a commitment scheme with statistical security, which means that a QU cannot claim two identical values of y_1, y_2 ($y_1 = y_2$) unless it can factorize h, solve the discrete logarithm of m in base n, or solve the discrete logarithm of n in base m. In short, assuming the factorization assumption, it is computationally infeasible to compute $\{y_1, y_2, r_1, r_1\}$ such that $Com_1(y_1, r_1) = Com_1(y_2, r_2)$.

In summary, the secret parameters $P(y, r)$ satisfying $Com_1 = m_y n_r \bmod$ cannot be obtained by the attacker when accessing sequence $O_{y,r}$ and commitment $\{Com_1, Com_2\}$. Therefore, the likelihood of the attacker being able to generate a valid signature $\lambda = PUB(Ks, c, I, I_1, I_2, Com_1, Com_2)$ is negligible. So, QU authenticity is assured.

Based on the elliptic curve discrete logarithm problem, the probability is negligible for the C_A and the C_B to generate their privacy keys SK_{H_1} and SK_{H_2} based on PK_{H_1} and PK_{H_2} and Com_{1_H}. Therefore, the authenticity of the C_A and the C_B is guaranteed. □

Theorem 3. *Ensuring the anonymity of the QU is a fundamental requirement of our solution, as it prohibits unauthorized access to personal information, rendering the attacker unable to retrieve identities.*

Proof. According to the FOC protocol, it can be observed that Com_1 and Com_2 provide no statistical information to the C_A and the C_B. In the process of mutual authentication, the QU generates a proof $\{c, I, I_1, I_2\}$ that can be verified by the C_A and the C_B without disclosing the actual identity. Additionally, attackers with access to the commitments cannot deduce the true identity of the QU, including in insider attacks. Hence, achieving user anonymity is possible. □

Theorem 4. *An attacker cannot authenticate by replaying legitimate information acquired during a query.*

Proof. In our proposed scheme, the random numbers $\{N_C, r_1, r_2\}$ are employed exclusively for the current query information, making it arduous for the attacker to acquire the preceding value. Additionally, the random numbers change with each set of query information, guaranteeing the freshness of each secret in the current query information. So, it is challenging to acquire any past query information. □

Theorem 5. *An attacker cannot authenticate by replaying legitimate information obtained from query information.*

Proof. Two scenarios need to be considered. In our solution, if an attacker attempts to replay an old message $\lambda = Enc_{PUB}(K_d, c, I, I_1, I_2)$ for authentication, it must satisfy one of the following two conditions:

$$N_C = N_C^{'} \tag{24}$$

$$c = c^{'}, c = M(W_1 \parallel W_2 \parallel N_C), c^{'} = H(W_1 \parallel W_2 \parallel N_C^{'}) \tag{25}$$

N_C is a cloud-selected replay random number.

For the condition $N_C = N_C^{'}$, due to the selection of C_A and C_B into a wide range of data N_C, the probability $Pr[N_C = N_C^{'}]$ can be neglected. The hash function selected by

the QU is assumed to possess ideal collision resistance, which means that the attacker is unable to produce an identical hash value without knowledge of N_C, given the condition $c = c^{'}, c = H(W_1 \parallel W_2 \parallel N_C), c^{'} = H(W_1 \parallel W_2 \parallel N_C^{'})$. Hence, our authentication is resistant to replay attacks. □

Theorem 6. *An attacker cannot obtain the true identity of users from anonymous authentication messages.*

Proof. ω, q_1, q_2 are random in the calculation of I, I_1, I_2, so proof of c, I, I_1, I_2 is sent to the C_A, and the C_B are different each time. The calculation process for $W_1^{'}, W_2^{'}$ is also different when C_A and the C_B perform the calculation. As a result, each access is associated with a unique and newly generated anonymous identity credential. Even if the attacker intercepts messages for an extended period, they will be unable to acquire any valuable information that could reveal the real identity of QU. Hence, the attacker cannot extract users' genuine identities from the anonymous authentication messages. □

6.3. Data Interaction between Clouds Security

Definition 3. *Let $H \subset P$ be the self of honest participants in set P. Assume that protocol Π can securely realize the functionality of F if there exists a polynomial time (PPT) set $Sim = (Sim_{P_1}, Sim_{P_2}, Sim_{C_B})$ such that, for all semi-honest PPT attackers A, their inputs are D_x, D_y and auxiliary input is z. For all participating parties, denoted as $p^* \in P$,*

$$REAL^{p^*}_{\pi,A,H,z}(\xi, x, y)_{\xi \in N} \equiv IDEAL^{p^*}_{F,Sim,H,z}(\xi, x, y)_{\xi \in N} \tag{26}$$

where $\equiv$ represents computational indistinguishability.

Theorem 7. *The secure computation protocol based on garbled circuits is secure when, at most, one malicious participant is present.*

Proof. Consider C_B as a malicious participant and C_A as a semi-honest participant. It must be demonstrated that the secure computation protocol achieves indistinguishability between the ideal and actual models. In other words, it is impossible to differentiate between each participant's interaction information and outputs in the ideal model and the accurate model during the following interactions.

In a real-world model, it is assumed that there exists a simulator that can simulate the various behaviors of a semi-honest participant C_A and receive inputs $(x_1, t_1), (x_2, t_2)$ from C_{A_1} and C_{A_2} from the protocol execution environment. Simultaneously, the simulator can mimic the functionality of the generating function F_f by forwarding all inputs to the simulated F_f. From the execution environment's standpoint, there is no distinction between the actual F_f and the simulated F_f since the simulator does not perform any computations carried out by F_f.

Because in Step 2, C_{A_1} and C_{A_2} uniformly selected the pseudo-random function seed r, it can be seen from the safety of the pseudo-random function that in Step 2, the actual model and the ideal model are indistinguishable.

In Step 3, we modify the simulator so that when it generates a commitment, it can know which commitments will be opened in advance. Firstly, the simulator can tag the randomly generated number O_1, O_2 corresponding to which commitment is opened and compute $a_1 = O_1 \oplus x_1 \parallel x_2$, $a_2 = O_2 \oplus t_1 \parallel t_2$. Meanwhile, the simulator obtains numerical values for (x_1, x_2) and (t_1, t_2). Then, the simulator can submit tag values that ensure they will remain unopened. In this process, as a result of promises being concealed, it becomes equally impossible to differentiate between the real and ideal models.

In Step 6, the simulated C_{A_1} and C_{A_2} cease execution when $De(r, Y^{'}) = 1$ is reached. Appropriate modifications are made to the simulator to achieve $Y^{'} \neq Ev(F, X)$. Authenticity through the garbled circuit C_B achieves $Y^{'} \neq Ev(F, X)$ only when $De(r, Y^{'}) = 1$,

with a negligible probability. Hence, it is equally impossible to distinguish between the implementation and the ideal models at this stage.

In Step 6, due to the correctness of the garbled circuit, both analog C_{A_1} and C_{A_2} can obtain outputs. Consequently, assuming no disruption occurred in the preceding phase, the simulator can be altered into an analog garbled circuit capable of producing (F, X, r). By simulating the instructions of F_f, we can replicate the output of both C_{A_1} and C_{A_2}. In this step, the indistinguishability between the real and ideal models is equally ensured based on the security of the garbled circuit.

In summary, the protocol ensures that the execution environment cannot differentiate between genuine and ideal models. As a result, the protocol remains secure even in the presence of a malicious participant represented by C_B. □

7. Experiment

In this section, an experimental analysis of the performance of our proposed scheme is conducted. We employed the Python programming language and the SEAL-Python library to implement SHE in the experiment. The experiments were performed in a Windows 10 environment with an Intel Core i5 2.30 GHz CPU and 16 GB RAM. In our experimental study, we compared our proposed scheme and the existing secure outsourcing schemes for KNN classification [18,19]. We primarily employed simulated data for experimental testing.

We conducted performance testing and analysis on three protocols: data outsourcing, data querying, and data interaction between clouds. Here, all experimental results are the average of 1000 test runs. Next, we provide a detailed analysis and explanation of the computational costs for each protocol.

7.1. Data Outsourcing

As shown in Figures 3 and 4, the execution time required for generating the encrypted synthetic data exhibits linear growth with increasing dimensionless d of the data nodes, and all have low computational overhead (as shown in Figure 3, it takes approximately 1.2 s for $d = 500$ and $n = 50$ K; in Figure 4, it takes around 0.56 s for $n = 1000$ K and $d = 20$). When encrypting the simulation data, compared to the existing schemes [18,19], our solution combining SHE and SIMD has a relatively low cost, so it has a certain degree of practicality.

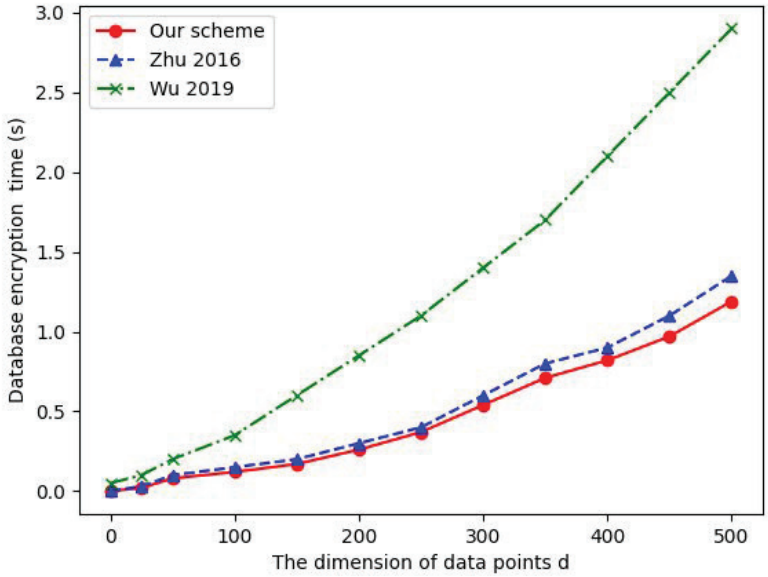

Figure 3. Average database encryption time (s) vs. d [18,19].

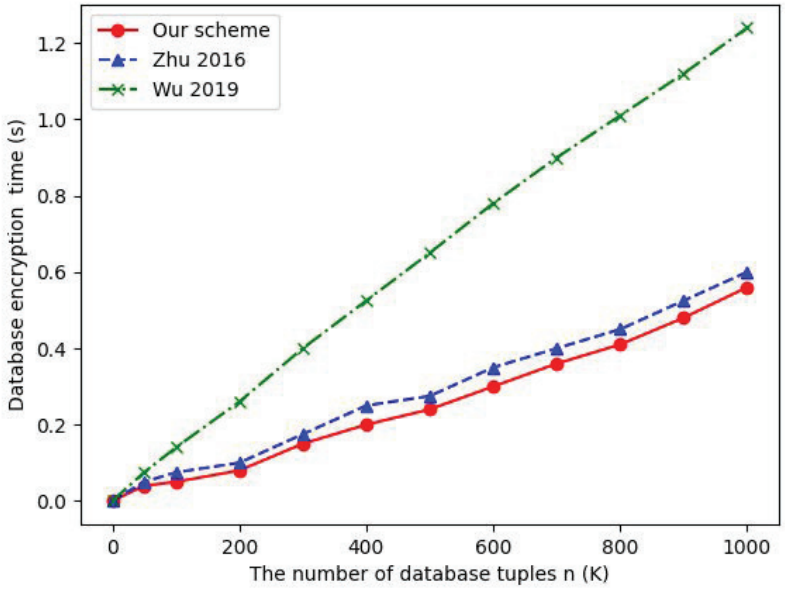

Figure 4. Average database encryption time (s) vs. n [18,19].

7.2. Data Query

Before performing query sample encryption, the QU needs to undergo anonymous mutual authentication based on FOC with C_A and C_B, which, although lightweight, still incur unavoidable computational overhead. Figure 5 demonstrates that our approach and the approach in [18] exhibit lower computational costs. Conversely, the costs of the approach in [19] rapidly escalate with an increase in the dimensionless data points, requiring approximately 120 s of running time for $d = 500$. In conclusion, our approach exhibits lower computational costs and superior practicality.

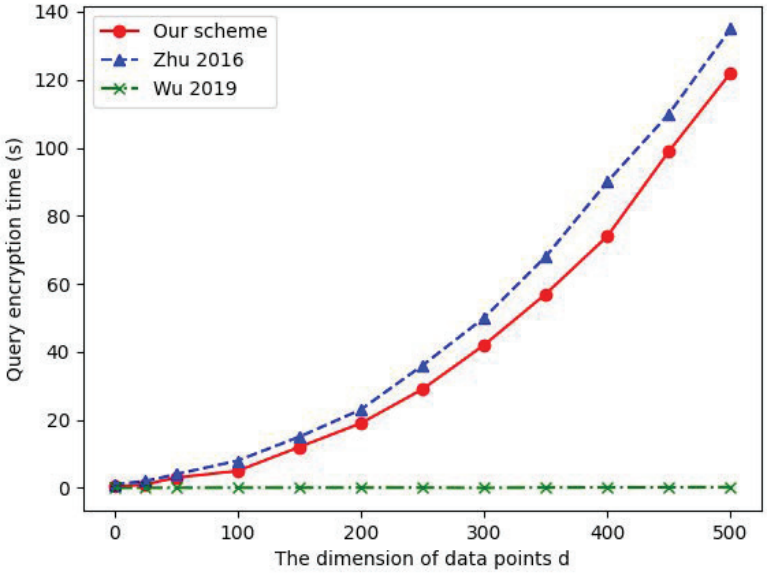

Figure 5. Average query encryption time (s) vs. d [18,19].

7.3. Data Interaction between Clouds

As shown in Figures 6 and 7, in our scheme, the running time is about 14 s when $d = 500$ and $n = 50$ K and approximately 150 s when $d = 20$ and $n = 1000$ K. Compared with existing schemes in [18,19], our approach incurs significant computational overhead in ciphertext KNN classification due to the requirement of executing secure averaging protocols based on garbled circuits between C_A and C_B, resulting in substantial time

consumption. The primary reason is the need for extensive bit commitment processing in the garbled circuit section; however, it can withstand attacks from a malicious cloud server in a dual-cloud environment, making it suitable for our proposed scheme. In future work, we will focus on improving performance concerning this issue.

Based on the comprehensive analysis of the experiments conducted, our approach requires additional computational overhead in data encryption, data querying, and data interaction between the cloud and the user due to potential malicious attackers controlling both the cloud and the QU. However, it should be noted that our solution still exhibits high computational efficiency while achieving secure KNN classification in the scenario where attackers control the cloud and the QU.

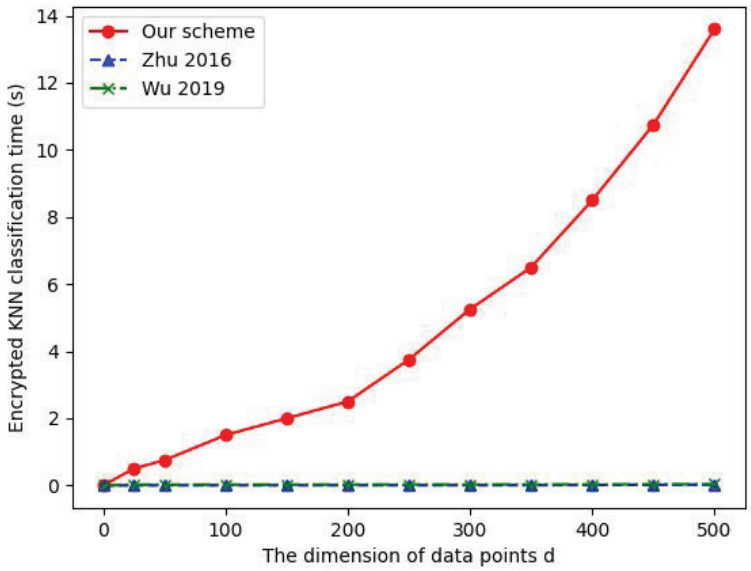

Figure 6. Average encrypted KNN classification time (s) vs. d [18,19].

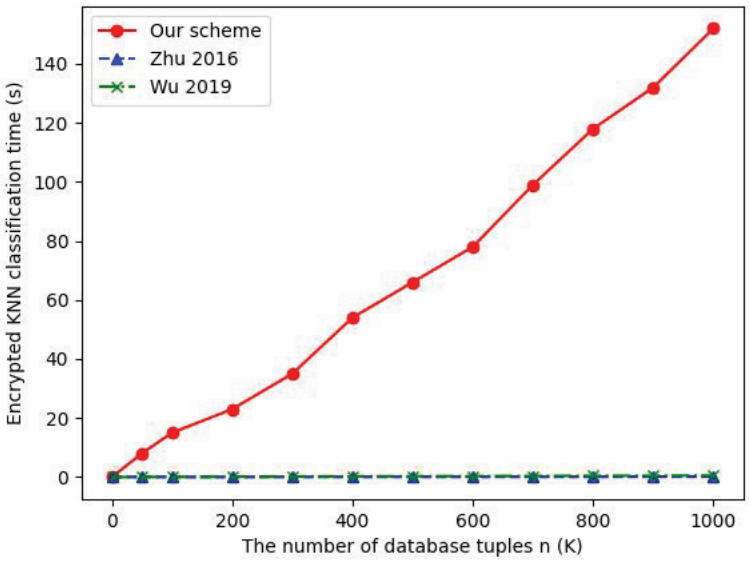

Figure 7. Average encrypted KNN classification time (s) vs. n [18,19].

8. Conclusions

This paper proposes a privacy-preserving KNN query scheme based on a secure multi-party computation mechanism to address security concerns when malicious attackers control the cloud and query users. We conducted a detailed security analysis of the proposed scheme, demonstrating its effectiveness in protecting the data privacy of the DO,

the QU, and the data interaction privacy between the two clouds. Finally, we evaluated its performance through experiments. The experimental results indicate that the scheme has a certain degree of feasibility and reliability.

In subsequent work, we will emphasize the balance between security and efficiency. Additionally, we will focus on practical applications in real-world scenarios.

Author Contributions: X.G., Y.L., Y.J., J.W., and J.F. contributed to the design and implementation of the research, to the analysis of the results, and to the writing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by NSFC no. 61461027, Gansu province science and technology plan under grant no. 20JR5RA467.

Data Availability Statement: No data were used to support this study.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

KNN	k-nearest neighbor
SHE	somewhat homomorphic encryption
SIMD	single-instruction multiple data
FOC	Fujisaki–Okamoto commitment
GEN	the generation algorithm
ENC	the encryption algorithm
DEC	the decryption algorithm
ECC	elliptic curve
DO	data owner
QU	query user

References

1. Shan, Z.; Ren, K.; Blanton, M.; Wang, C. Practical Secure Computation Outsourcing: A Survey. *Acm Comput. Surv. (CSUR)* **2018**, *51*, 1–40. [CrossRef]
2. Zhang, M.; Zhang, Y.; Shen, G. PPDDS: A privacy-preserving disease diagnosis scheme based on the secure Mahalanobis distance evaluation model. *IEEE Syst. J.* **2021**, *16*, 4552–4562. [CrossRef]
3. Zissis, D.; Lekkas, D. Addressing cloud computing security issues. *Future Gener. Comput. Syst.* **2012**, *28*, 583–592. [CrossRef]
4. Zhang, M.; Zhang, Y.; Jiang, Y.; Shen, J. Obfuscating EVES algorithm and its application in fair electronic transactions in public clouds. *IEEE Syst. J.* **2019**, *13*, 1478–1486. [CrossRef]
5. Barona, R.; Anita, E.A.M. A survey on data breach challenges in cloud computing security: Issues and threats. In Proceedings of the International Conference on Circuit, Power and Computing Technologies (ICCPCT), Bhubaneswar, India, 20–21 April 2017.
6. Zhang, M.; Chen, Y.; Lin, J. A privacy-preserving optimization of neighborhood-based recommendation for medical-aided diagnosis and treatment. *IEEE Internet Things J.* **2021**, *8*, 10830–10842. [CrossRef]
7. Zhang, M.; Chen, Y.; Lin, J. Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges. *Comput. Commun.* **2019**, *140*, 38–60.
8. Zhang, M.; Chen, Y.; Lin, J. A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *27*, 340–352.
9. Wang, B.; Liao, Q.; Zhang, C. Weight Based KNN Recommender System. In Proceedings of the 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 26–27 August 2013.
10. Barona, R.; Anita, E.A.M. *Privacy-Preserving Distributed k-Means Clustering over Arbitrarily Partitioned Data*; ACM SIGKDD: Chicago, IL, USA, 2005.
11. Fan, J.; Vercauteren, F. Somewhat Practical Fully Homomorphic Encryption. 2012. Available online: https://eprint.iacr.org/2012/144 (accessed on 8 November 2023).
12. Smart, N.P.; Vercauteren, F. Fully Homomorphic SIMD Operations. *Des. Codes Cryptogr.* **2014**, *71*, 57–81. [CrossRef]
13. Yiu, M.L.; Assent, I.; Jensen, C.S.; Kalnis, P. Outsourced Similarity Search on Metric Data Assets. *IEEE Trans. Knowl. Data Eng.* **2010**, *24*, 338–352. [CrossRef]
14. Boudot, F. Efficient proofs that a committed number lies in an interval. In Proceedings of the Advances in Cryptology—EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, France, 14–18 May 2000; Springer: Berlin/Heidelberg, Germany, 2000.

15. Barona, R.; Anita, E.A.M. An Authentication Scheme in VANETs Based on Group Signature. In Proceedings of the Intelligent Computing Theories and Application: 15th International Conference, Nanchang, China, 3–6 August 2019.
16. Wong, W.K.; Cheung, D.W.; Kao, B.; Mamoulis, N. Secure kNN Computation on Encrypted Databases. In Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, Providence, RI, USA, 29 June–2 July 2009.
17. Zhou, L.; Zhu, Y.; Castiglione, A. Efficient k-NN query over encrypted data in cloud with limited key-disclosure and offline data owner. *Comput. Secur.* **2017**, *69*, 84–96. [CrossRef]
18. Wu, W.; Parampalli, U.; Liu, J.; Xian, M. Privacy preserving k-nearest neighbor classification over encrypted database in outsourced cloud environments. *World Wide Web* **2019**, *22*, 101–123. [CrossRef]
19. Zhu, Y.; Huang, Z.; Takagi, T. Secure and controllable k-NN query over encrypted cloud data with key confidentiality. *J. Parallel Distrib. Comput.* **2016**, *89*, 1–12. [CrossRef]
20. Elmehdwi, Y.; Samanthula, B.K.; Jiang, W. Secure k-nearest neighbor query over encrypted data in outsourced environments. In Proceedings of the International Conference on Data Engineering, Chicago, IL, USA, 31 March–4 April 2014.
21. Guan, Y.; Lu, R.; Zheng, Y.; Shao, J.; Wei, G. Toward Oblivious Location-Based k-Nearest Neighbor Query in Smart Cities. *IEEE Internet Things J.* **2021**, *8*, 14219–14231. [CrossRef]
22. Samanthula, B.K.; Elmehdwi, Y.; Jiang, W. Privacy-Preserving k-Nearest Neighbor Computation in Multiple Cloud Environments. *IEEE Access* **2016**, *4*, 9589–9603.
23. Samanthula, B.K.; Elmehdwi, Y.; Jiang, W. k-Nearest Neighbor Classification over Semantically Secure Encrypted Relational Data. *IEEE Trans. Knowl. Data Eng.* **2014**, *27*, 1261–1273. [CrossRef]
24. Cui, N.; Yang, X.; Wang, B.; Li, J.; Wang, G. SVkNN: Efficient Secure and Verifiable k-Nearest Neighbor Query on the Cloud Platform. In Proceedings of the International Conference on Data Engineering (ICDE), Dallas, TX, USA, 20–24 April 2020.
25. Liu, Q.; Hao, Z.; Peng, Y.; Jiang, H.; Wu, J.; Peng, T.; Wang, G.; Zhang, S. SecVKQ: Secure and verifiable kNN queries in sensor–cloud systems. *IEEE Trans. Knowl. Data Eng.* **2021**, *120*, 102300. [CrossRef]
26. Yang, S.; Tang, S.; Zhang, X. Privacy-preserving k nearest neighbor query with authentication on road networks. *J. Parallel Distrib. Comput.* **2019**, *134*, 25–36. [CrossRef]
27. Wu, W.; Liu, J.; Rong, H.; Wang, H.; Xian, M. Efficient k-Nearest Neighbor Classification Over Semantically Secure Hybrid Encrypted Cloud Database. *IEEE Access* **2018**, *6*, 41771–41784. [CrossRef]
28. Lian, H.; Qiu, W.; Yan, D.; Huang, Z.; Tang, P. Efficient and secure k-nearest neighbor query on outsourced data. *Peer Netw. Appl.* **2020**, *13*, 2324–2333. [CrossRef]
29. Du, J.; Bian, F. A Privacy-Preserving and Efficient k-Nearest Neighbor Query and Classification Scheme Based on k-Dimensional Tree for Outsourced Data. *IEEE Access* **2020**, *8*, 69333–69345. [CrossRef]
30. Jiang, X.; Li, L. Efficient secure and verifiable KNN set similarity search over outsourced clouds. *High-Confid. Comput.* **2023**, *3*, 100100. [CrossRef]
31. Pei, X.; Li, L.; Jiang, X. Efficient privacy-preserving k-nearest neighbors in cloud computing. In Proceedings of the International Conference on Cloud Computing, Internet of Things, and Computer Applications (CICA 2022), Dubai, United Arab Emirates, 22 March 2022 .
32. Hsu, Y.C.; Hsueh, C.H.; Wu, J.L. A Privacy Preserving Cloud-Based K-NN Search Scheme with Lightweight User Loads. *Computers* **2020**, *9*, 1. [CrossRef]
33. Zheng, Y.; Lu, R.; Guan, Y.; Shao, J.; Zhu, H. Achieving Efficient and Privacy-Preserving Exact Set Similarity Search over Encrypted Data. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 1090–1103. [CrossRef]
34. Zuber, M.; Sirdey, R. Efficient homomorphic evaluation of k-NN classifiers. *Enhancing Technol.* **2021**, *2021*, 111–129. [CrossRef]
35. Li, Z.; Tian, G.; Tan, S. Secure and Efficient k-Nearest Neighbor Query with Privacy-Preserving Authentication. In Proceedings of the International Symposium on Security and Privacy in Social Networks and Big Data, Hangzhou, China, 22 July 2022 .
36. Ameur, Y.; Aziz, R.; Audigier, V.; Bouzefrane, S. Secure and Non-interactive k-NN Classifier Using Symmetric Fully Homomorphic Encryption. In Proceedings of the International Conference on Privacy in Statistical Databases, Paris, France, 21 September 2022 .
37. Cheng, K.; Wang, L.; Shen, Y.; Wang, H.; Wang, Y.; Jiang, X.; Zhong, H. Secure k-NN Query on Encrypted Cloud Data with Multiple Keys. *IEEE Trans. Big Data* **2017**, *7*, 689–702. [CrossRef]
38. Cui, N.; Qian, K.; Cai, T.; Li, J.; Yang, X.; Cui, J.; Zhong, H. Towards Multi-User, Secure, and Verifiable k-NN Query in Cloud Database. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 9333–9349. [CrossRef]
39. Naor, M. Bit Commitment Using Pseudorandomness. *Des. Codes Cryptogr.* **1991**, *4*, 151–158. [CrossRef]
40. Blum, M.; Feldman, P.; Micali, S. Non-Interactive Zero-Knowledge and Its Applications. In Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, 2–4 May 2019.
41. Kim, J.; Koo, D.; Kim, Y.; Yoon, H.; Shin, J.; Kim, S. Efficient Privacy-Preserving Matrix Factorization for Recommendation via Fully Homomorphic Encryption. *Des. Codes Cryptogr.* **2014**, *71*, 57–81. [CrossRef]