Privacy-Preserving Decision Tree Classification Protocol Based on Bitwise Comparison

Peihang Yu[1(✉)], Baodong Qin[1], and Dong Zheng[2]

[1] School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
19991009586@163.com, qinbaodong@xupt.edu.cn
[2] School of Computer Qinghai Normal University, Xining 810008, China

Abstract. Decision tree model is widely used in telemedicine, credit evaluation and other fields because of its high efficiency and ease of use. Companies can charge customers who do not have professional knowledge or resources to build pre-diction models to achieve the purpose of profitability. In order to protect the privacy of client data and decision tree model parameters in this scenario, we propose an integer comparison protocol based on distributed bitwise comparison, and further design an efficient privacy-preserving decision tree classification model. Our protocol uses the idea of Beaver triple multiplication combined with secret sharing to replace the ciphertext homomorphic operation in the original comparison protocol. Compared with the comparison protocol relying on the traditional semi-homomorphic encryption scheme, it further im-proves the operation efficiency and compresses the communication cost in ciphertext transmission. Security analysis shows that this scheme achieves the expected privacy for users and model providers. Experimental results on real datasets show that the overall running time of this scheme is about 0.8s for a decision tree of depth 3, and about 5s for a decision tree of depth 17. So, its computational overhead is low and acceptable.

Keywords: Machine learning · Decision tree · Secret sharing · Homomorphic encryption · Privacy-preserving

1 Introduction

With the diversification of Internet technology more and more companies are offering a wide range of remote services to their users. These include machine learning classifiers for data analysis and prediction services such as healthcare, transportation, and image recognition. In actual deployment, typically the classification service provider has a trained model, the user gives an instance as

Y. Chenggang et al. (Eds.): MobiMedia 2022, LNICST 451, pp. 91–104, 2022.
https://doi.org/10.1007/978-3-031-23902-1_8

input, and its output represents the classification label of its output. The classifier is trained by the hospital with human and material resources, which is a trade secret and cannot be disclosed [1]. The user's input includes their own, such as height, weight, heart rate, etc., which are personal privacy which cannot be disclosed. Therefore, privacy protection in the use of machine learning models is crucial. Since Agrawal et al. [2] proposed privacy-preserving data mining techniques, mainstream data mining techniques have emerged successively with solutions to privacy-preserving problems such as neural networks [3], support vector machines [4], decision trees [5] and random forests [6], etc. In this paper, we focus on privacy-preserving issues related to decision tree models, its related work is as follows.

In 2015, Bost et al. [7] were the first to design a privacy-preserving decision tree evaluation scheme using FHE technique, which uses FHE to encrypt the data and process the decision tree as a polynomial, and computes the polynomial by homomorphism to finally obtain the classification result. Subsequently, Wu et al. [8] used a more lightweight additive homomorphic encryption combined with oblivious transfer(OT) [9] instead of fully homomorphic encryption to reduce the computational overhead. Tai et al. [10] proposed to represent the decision tree as multiple linear functions and determine whether a single leaf node contains a classification result by calculating the path cost of that leaf node, which reduces the computational overhead and communication compared to the Wu [8]. In 2019, Zheng et al. [11] proposed to use additive secret sharing technique to divide the decision tree thresholds with user features into two secret shares and outsource to two servers to complete the overall classification process using MSB, but requires a large number of parameters to be prepared in advance by trusted third parties. In 2020, Xue et al. [12] used Paillier encryption scheme to encrypt the data using respective shares to do the difference comparison size. Recently, Cao et al. [13] proposed a multi-key privacy preserving decision tree evaluation scheme based on double trapdoor encryption scheme, the scheme encrypts data using DT-PKC encryption system with additive homomorphic property and uses DGK [14] comparison protocol for node comparison, the overall efficiency of the scheme is low due to the low efficiency of double trapdoor encryption scheme and the per-bit comparison of decision nodes.

Our Contribution. In this paper, we propose a privacy-preserving decision tree classification protocol in the client-server model. The protocol improves the bit-by-bit comparison scheme proposed by Garay et al. [15] to compare user's features with decision nodes bit-by-bit, while ensuring the confidentiality of user input and decision tree thresholds. It uses Beaver multiplicative triples to complete the multiplicative operations in the comparison protocol, and avoids the bit encryption by homomorphic encryption scheme. Hence, it improves the efficiency of the comparison protocol in the decision tree classification protocol. To generate the final classification result, the decision tree is transformed into multiple linear functions and combined with Paillier homomorphic encryption to complete the cipher-text operation and selection of classification results.

The rest of this paper is organized as follows: In Sect. 2, we recall the definition of decision tree classification and some cryptographic tools, including Beaver multiplicative triple and Paillier cryptosystem. In Sect. 3, we introduce the system model and its security model. Our main protocol and its security analysis are given in Sect. 4 and Sect. 5 respectively. We evaluate its performance in Sect. 6. Finally, we conclude this paper in Sect. 7.

2 Preliminaries

2.1 Decision Tree Classification

Assume that the input to the user side of the decision tree model is in the form of an n dimensional feature vector $X = (x_1, \ldots, x_n) \in Z^n$, the number of internal nodes of the decision tree is m, and the node threshold vector is $Y = (y_1, \ldots, y_m) \in Z^m$. We assume that the decision tree is a complete binary tree and each node has 0 or 2 children, as shown in Fig. 1. A complete binary tree with m non-leaf nodes has $m + 1$ leaf nodes and the leaf nodes are denoted by $V = (v_1, v_2, \ldots, v_{m+1})$, then the evaluation function of the decision tree is denoted as $T : Z^n \rightarrow \{v_1, v_2, \ldots, v_{m+1}\}$. The output of the decision tree is denoted as $v = T(X)$, v representing the classification result to which the input X belongs. The decision process of a decision tree starts at the root node, tests the conditions on the current node and descends to the left branch node or right branch node until it reaches some leaf node storing $T(X)$. Usually, each decision point corresponds to a Boolean function $f_k(X) = 1\{x_{i_k} > y_k\}$, where $k \in \{1, 2, \ldots, m\}$ and $i_k \in \{1, 2, \ldots, n\}$. If and only if x_{i_k} is greater than y_k, then $f_k(X)$ equals to 1. When $f_k(X) = 1$, then the next step goes to the left branch of the decision node; Otherwise, it goes to the right branch of the node.

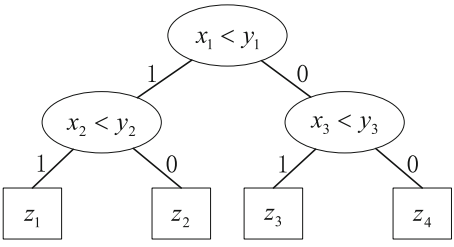

Fig. 1. An example of decision tree.

2.2 Beaver Multiplicative Triple

The Beaver triplet was proposed by Donald Beaver [16] and is mainly applied to multiplication operations in secure multi-party computation protocols. To simplify notation, we use "[·]" denote the secret sharing of "·".

Suppose the communicating parties A and B have the secret sharing of x and y. Party A has share $[x]_A$ and $[y]_A$, and Party B has share $[x]_B$ and $[y]_B$. They satisfy $[x]_A + [x]_B = x$ and $[y]_A + [y]_B = y$. Now, the secret share $[xy]$ can be calculated by Party A and Party B according to the following steps:

1. A trusted third party generates a multiplicative triple (u, g, z), which satisfies $z = u \cdot g$. The triple is kept secret from the two parties in the operation, but each of the two parties has the secret share $[z]$, $[u]$ and $[g]$.
2. Party A and Party B calculate the shares of $\alpha = x - u$ respectively, and made them public, i.e., $[\alpha]_A = [x]_A - [u]_A$ and $[\alpha]_B = [x]_B - [u]_B$.
3. Party A and Party B calculate the shares of $\beta = y - g$ respectively, and made them public, i.e., $[\beta]_A = [y]_A - [g]_A$ and $[\beta]_B = [y]_B - [y]_B$.
4. The two parties reconstruct α and β according to the secret shares.
5. Party A and Party B calculate the shares of η, that is

$$[\eta]_A = [z]_A + \alpha \cdot [g]_A + \beta \cdot [u]_A \text{ and } [\eta]_B = [z]_B + \alpha \cdot [g]_B + \beta \cdot [u]_B + \alpha \cdot \beta.$$

The correctness of above steps to calculate the shares $[xy]$ is showed as follows: Since the shares $[\alpha]$ and $[\beta]$ are public, each party can reconstruct α and β, and hence the above shares $[\eta]_A$ and $[\eta]_B$. So,

$$\begin{aligned}[\eta]_A + [\eta]_B &= [z]_A + \alpha \cdot [g]_A + \beta \cdot [u]_A + [z]_B + \alpha \cdot [g]_B + \beta \cdot [u]_B + \alpha \cdot \beta. \\ &= u \cdot g + (x - u) \cdot g + (y - g) \cdot u + (x - u) \cdot (y - g) \\ &= x \cdot y.\end{aligned}$$

2.3 Paillier Encryption Algorithm

The Paillier encryption algorithm [16] is a public key encryption scheme based on the composite residuosity problem, proposed by Paillier et al. in 1999. It supports additive homomorphic operations over the message space Z_N and is widely used in various security domains. In this paper, we use $\|\cdot\|$ to denote the encryption of "$\cdot$", and use $\mathsf{Dec}(\cdot)$ to denote the decryption of C. Assume that m_1 and m_2 are two plaintext messages, and r_1 and r_2 are two random numbers in Z_N^*. (N, g) is the public key of the Paillier encryption algorithm, where N is the product of two large prime numbers p, q and $g = 1 + N$. The Paillier encryption algorithm is additively homomorphic, as it has the following two properties.

1. **Addition:** For any two messages $m_1, m_2 \in Z_N$, it has that

$$\begin{aligned}\|m_1\| \cdot \|m_2\| &= \left(g^{m_1} r_1^N \bmod N^2\right) \cdot \left(g^{m_2} r_2^N \bmod N^2\right) \\ &= g^{m_1+m_2} (r_1 r_2)^N \bmod N^2 \\ &= \|m_1 + m_2\|.\end{aligned}$$

2. **Scalar Multiplication:** For any integer $k \in Z_N$, it has that

$$\begin{aligned}\|m_1\|^k &= \left(g^{m_1} r_1^N \bmod N^2\right)^k \\ &= g^{k \cdot m_1} (r_1^k)^N \bmod N^2 \\ &= \|k \cdot m_1\|.\end{aligned}$$

3 System Model and Security Model

3.1 System Model

The system involves three main entities: a user, a service provider, and a trusted authority. The specific interactions are shown in the following Fig. 2.

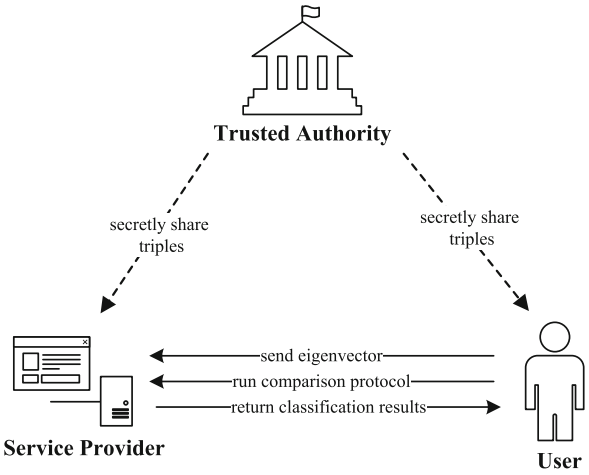

Fig. 2. System model.

- **User (U).** The user has a private input in the form of a feature vector, which contains information about a different attribute of the user, such as height, weight, heart rate etc. During the operation of the protocol, the user wants to use the decision tree model to obtain the classification results corresponding to their input. As the feature vector may contain sensitive personal information, the user will not send data in plain text during the operation of the protocol.
- **Service provider (P).** This entity is the party in the protocol that work the major computational overhead, usually a commercial company, hospital, etc. It holds a decision tree model that has been trained by a decision tree algorithm (e.g. CART, C4.5, etc.) and uses it to provide a classification service to the user. During the classification process, the model parameters should be avoided to be leaked to users during the protocol process while ensuring the decision tree classification function.
- **Trust Authority (TA).** This entity performs the operations related to Beaver tri-ples. At the beginning of the protocol, it first generates the corresponding number of triples and shares them secretly with the user and the classification service pro-vider for the subsequent multiplication of the triples.

3.2 Security Model

Unlike some research on privacy preservation in the training phase [18–20], we mainly focus on the existence of security issues in the process of decision tree.

We use U and P to denote the user and the service provider, respectively. De-note by A an attacker that can collude with either the user or the service provider to obtain the secret information of the other party. Let $X = (x_1, x_2, \ldots, x_m) \in Z^m$ denote the user's feature vector. $Y = (y_1, y_2, \ldots, y_m) \in Z^m$ and $Z = (z_1, z_2, \ldots, z_n)$ denote the node threshold vector and the classification result vector of the decision tree, respectively. Let $U(X) \leftrightarrow P(Y, Z)$ denote the running process of the protocol, and $(U(X) \leftrightarrow P(Y, Z)) = v$ denote the classification results obtained by the user. The privacy of the user and the privacy of the decision tree model can be defined respectively as follows.

- **Privacy of User.** When A colludes with the model holder P, A can obtain the model parameters Y and Z. When a user initiates a protocol $U(X) \leftrightarrow P(Y, Z)$, the attacker can obtain all the information from the user during the execution of the protocol. The attacker should not be able to obtain other information about the user's feature vectors $X = (x_1, x_2, \ldots, x_m)$ and classification results.
- **Privacy of Model.** When attacker A conspires with U, the attacker can choose a different feature vector $X_j = (x_{j,1}, x_{j,2}, \ldots, x_{j,m})$, initiate protocol $U(X_j) \leftrightarrow P(Y, Z)$. The attacker can obtain all the information from the service provider during the protocol execution and the corresponding classification result v_j. Except for the dimensionality of the node threshold vector, the dimensionality of the classification result vector and the classification result v_j, The attacker should not be able to obtain the node threshold vector and other information about the classification result vector.

4 Privacy-Preserving Decision Tree Classification

4.1 Secure Decision Node Comparison

This section focuses on the secure comparison of the values owned by the participating service provider and users of the protocol. It uses additive secret sharing and comparison scheme of [17], and allows the service provider to obtain the encrypted comparison results for each decision node.

Suppose a feature value of the user is represented in bits $c_l \cdots c_2 c_1$, and a node threshold y_p of the decision tree model is represented in bits $p_l \cdots p_2 p_1$. That is $x_c = \sum_{k=1}^{l} c_k 2^{k-1}$ and $y_p = \sum_{k=1}^{l} p_k 2^{k-1}$, where l is the bit length of x_c and y_p. Let t_i denote the result of comparing $\sum_{k=1}^{i} c_k 2^{k-1}$ with $\sum_{k=1}^{i} p_k 2^{k-1}$, i.e., the result of comparing the first i bits of x_c with the first i bits of y_p. When $\sum_{k=1}^{i} c_k 2^{k-1} > \sum_{k=1}^{i} p_k 2^{k-1}$, t_i equals to 1; Otherwise, it is 0. To obtain the result of the comparison between x_c and y_p, it is necessary to compare the iterative formula of the protocol to calculate:

$$t_i = (1 - (c_i - p_i)^2) t_{i-1} + c_i (1 - p_i) \tag{1}$$

where i denotes the number of bits and $1 \leq i \leq l$, $t_0 = 0$. When i equals to l, t_l is the final comparison result. Under the modulo 2 operations, the formula Eq. 1 is further simplified into the following equation:

$$t_i = (1 + c_i + p_i)t_{i-1} + c_i(1 + p_i) \tag{2}$$

As both the provider and the user are honest and curious, both parties can secretly save each other's bits during the iterative computation process of t_l, and thus recover the full threshold and user's feature values. To solve this problem, we use additive secret sharing combined with Beaver triples to solve the multiplication operation in Eq. 2, including $(1 + c_i + p_i)t_{i-1}$ and $c_i(1 + p_i)$.

In detail, for each $i \in 1, 2, \ldots, l$, the user side and the service provider divide each bit of secret sharing x_c with y_p into $[c_i]_0$, $[c_i]_1$, $[p_i]_0$ and $[p_i]_1$, where the subscript 0 indicates the secret share of the user and the subscript 1 indicates the secret share of the service provider. Specifically,

$$[c_i]_0 = c_i,\ [c_i]_1 = 0,\ [p_i]_0 = 0 \text{ and } [p_i]_1 = p_i$$

According to the above approach, the user and the provider can calculate the secret sharing of $1 + c_i + p_i$ and $1 + p_i$ respectively:

$$\begin{aligned} [1 + c_i + p_i]_0 &= 1 + [c_i]_0 + [p_i]_0 & \qquad [1 + p_i]_0 &= 1 + [p_i]_0 \\ [1 + c_i + p_i]_1 &= [c_i]_1 + [p_i]_1 & \qquad [1 + p_i]_1 &= [p_i]_1 \end{aligned}$$

Initializing the secret sharing of t_0 as $[t_0]_0 = [t_0]_1 = 0$. Using the prepared Beaver triple for multiplication, the user and the provider calculate the secret sharing of t_i respectively. With the help of the Beaver triple, they compute the secret shares of $(1 + c_i + p_i)t_{i-1}$ and $c_i(1 + p_i)$ respectively. Then, the additive secret sharing of t_i is calculated for both parties as:

$$\begin{aligned} [t_i]_0 &= [(1 + c_i + p_i)t_{i-1}]_0 + [c_i(1 + p_i)]_0 \\ [t_i]_1 &= [(1 + c_i + p_i)t_{i-1}]_1 + [c_i(1 + p_i)]_1 \end{aligned}$$

The process is iterated until i equals to l. Finally, the user encrypts $[t_l]_0$ using the Paillier public key and sends $\|[t_l]_0\|$ to the provider. Then, the provider calculates $\|t_l\|$ based on its own secret sharing:

- If $[t_l]_1 = 0$, then $\|t_l\| = \|[t_l]_0\|$;
- If $[t_l]_1 = 1$, then $\|t_l\| = \|1 - [t_l]_0\|$.

For each threshold y_k in the node threshold vector of the decision tree model $Y = (y_1, y_2, \ldots, y_m)$ and the corresponding feature value x_{i_k} in the feature vector $X = (x_1, y_2, \ldots, x_n)$, the user runs the protocol and the service provider can then calculate the ciphertexts of the comparison results for any decision node $\|B\| = \{\|b_1\|, \|b_2\|, \ldots, \|b_m\|\}$, where $\|b_j\|$ is the ciphertext of the comparison results for the j-th decision node. More details are shown in Protocol 1.

Protocol 1: Secure Decision Node Comparison

Input (U): x_c, pk
Input (P): y_p, pk
Output (P): $\|b\|$
1. U & P:
(1) x_c and y_p are represented in bits, i.e., $x_c = c_l \cdots c_1$, $y_p = p_l \cdots p_1$.
(2) Define the secret sharing of each bit:
$[c_i]_0 = c_i$, $[p_i]_0 = 0$, $[c_i]_1 = 0$, $[p_i]_1 = p_i$, $i \in \{1, 2, \ldots, l\}$
(3) Define secret sharing according to step (2):
$[1 + c_i + p_i]_0 = 1 + [c_i]_0 + [p_i]_0$, $[1 + p_i]_0 = 1 + [p_i]_0$
$[1 + c_i + p_i]_1 = [c_i]_1 + [p_i]_1$, $[1 + p_i]_1 = [p_i]_1$
2. U & P:
(1) Set $[t_0]_0 = [t_0]_1 = 0$, then calculate the value of $1 + c_i + p_i$ and $1 + p_i$ using Beaver triple.
(2) Calculate the secret shares of t_i:
$[t_i]_0 = [(1 + c_i + p_i)t_{i-1}]_0 + [c_i(1 + p_i)]_0$
$[t_i]_1 = [(1 + c_i + p_i)t_{i-1}]_1 + [c_i(1 + p_i)]_1$
z (3) Repeat steps (1)(2) until $i = l$.
3. U:
Encrypt $[t_l]_0$ using pk, then send the ciphertext $\|[t_l]_0\|$ to P.
4. P:
Calculate the ciphertext $\|t_l\|$ according to $[t_l]_1$:

$$\|t_l\| = \begin{cases} \|[t_l]_0\| & \text{if } [t_l]_1 = 0 \\ \|1 - [t_l]_0\| = \|1\| \cdot \|[t_l]_0\|^{N-1} & \text{if } [t_l]_1 = 1 \end{cases}$$

Denote $\|t_l\|$ as the ciphertext of decision node comparison result $\|b\|$.

4.2 Secure Path Evaluation

After running the comparison protocol of the previous section for each node of the decision tree, the classification service provider obtains the encrypted comparison result $\|b_j\|$ for each decision node $Node_j$ and the set of encrypted comparison results are denoted as $\|B\| = \{\|b_1\|, \ldots, \|b_j\|, \ldots, \|b_m\|\}$. In this section, the classification service provider will perform path evaluation using $\|B\|$ to obtain the path cost of all leaf nodes.

In detail, for each decision node $Node_j$ associated with its two branches: the left branch is assigned edge cost $e_{j,left} = b_j$ and the right branch is assigned edge cost $e_{j,right} = 1 - b_j$. The path cost P_k for each leaf node is defined as the sum of all edge costs on the path to that leaf node, and the classification value v_k contained classification result of the decision tree if and only if $P_k = 0$.

Combined with the path cost mechanism in [13], the secure path evaluation of this scheme is shown in Protocol 2. First, the classification service provider sets the left edge cost of each decision node to be $\|e_{j,left}\| = \|b_j\|$. For the right edge cost, it is $\|e_{j,right}\| = \|1\| \cdot \|b_j\|^{N-1}$ using the homomorphic property of Paillier cryptosystem. Finally, the classification service provider multiplies the ciphertext of the edge cost on each leaf node path to obtain the ciphertexts $\|P\| = \{\|P_1\|, \|P_2\|, \ldots, \|P_{m+1}\|\}$ of each leaf node path cost.

Protocol 2: Secure Path Evaluation

Input (P): $\|b_j\|$, pk
Output (P): $\|P_k\|$
1. P:
For each decision node $Node_j$, set:
$\|e_{j,left}\| = \|b_j\|$ and $\|e_{j,right}\| = \|1 - b_j\| = \|1\| \cdot \|b_j\|^{N-1}$.
2. P:
Set the path cost of $Leaf_k$: $\|P_k\|$ is the concatenated product of each edge cost $\|e\|$ on that path.

4.3 Secure Classification Result Generation

After the secure path evaluation phase, the service provider obtains the ciphertext of the path cost of each leaf node. In this phase, the provider outputs the corresponding classification results based on the ciphertext of the path cost of the leaf nodes and sends them to the user. The user decrypts the ciphertexts to obtain the final classification results.

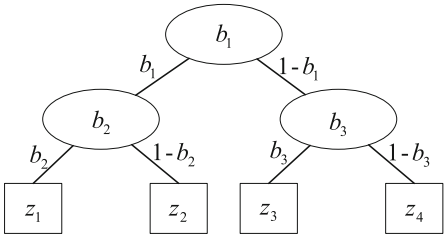

Fig. 3. System model.

Suppose the current decision tree has $m + 1$ leaf nodes, and denote by $Leaves = \{leaf_1, leaf_2, \ldots, leaf_{m+1}\}$ the set of leaf nodes. Each leaf node has the corresponding classification result $V = \{v_1, v_2, \ldots, v_{m+1}\}$ and its corresponding path cost $P = \{P_1, P_2, \ldots, P_{m+1}\}$. The protocol first transforms the decision tree model into $m+1$ linear functions according to equation $h_k = P_k + v_k$, where $k \in \{1, 2, \ldots, m + 1\}$. When the path cost $P_k = 0$, the value of h_k is the final classification result, and the user can use this property to recover the correct classification result v_k. As shown in Fig. 3, the linear function of each leaf node of the decision tree is defined as follows:

$$P_1 = b_1 + b_2 \qquad h_1 = P_1 + v_1$$
$$P_2 = b_1 + (1 - b_2) \qquad h_2 = P_2 + v_2$$
$$P_3 = 1 - b_1 + b_3 \qquad h_3 = P_3 + v_3$$
$$P_4 = 1 - b_1 + (1 - b_3) \qquad h_4 = P_4 + v_4$$

In detail, the protocol is shown in Protocol 3 below. First, for each $k \in \{1, 2, \ldots, m+1\}$, it selects a random number $r_k \in Z_P^*$ and multiples it with the path cost P to obtain $P_k^* = \|P_k\|^{r_k}$ by the Paillier encryption homomorphic property. This step is a randomized masking of the path cost and classification values of the leaf nodes. It ensures that the user will only recover the right classification value after receiving them and hence achieves the purpose of hiding the decision tree structure from the user side.

Protocol 3: Secure Classification Result Generation

Input (P): $\|P\| = \{\|P_1\|, \|P_2\|, \ldots, \|P_{m+1}\|\}$, pk
Output (U): v
1. P:
For each leaf node $leaf_k$, where $k \in \{1, 2, \ldots, m+1\}$, do:
(1) Randomly select $r_k \in Z_P^*$.
(2) Calculate $P_k^* = \|r_k \cdot P_k\| = \|P_k\|^{r_k}$.
(3) Encrypt v_k, denoted as $\|v_k\|$.
(4) Calculate $h_k^* = P_k^* \cdot \|v_k\| = \|r_k \cdot P_k + v_k\|$.
Send $H^* = \{h_1^*, h_2^*, \ldots, h_{m+1}^*\}$ and $\{P_1^*, P_2^*, \ldots, P_{m+1}^*\}$ to U.
2. U:
(1) For every $k \in \{1, 2, \ldots, m+1\}$, decrypt the P_k^* corresponding to each leaf node.
(2) If $\mathsf{Dec}(P_k^*) = 0$, decrypt the corresponding h_k^* to be the classification result v.

The service provider then sends to the user these randomized linear functions $H^* = \{h_1^*, h_2^*, ..., h_{m+1}^*\}$ and P_k^* corresponding to each h_k^*, where $h_k^* = \|r_k \cdot P_k^* + v_k\|$ and $k \in \{1, 2, \ldots, m+1\}$. The user decrypts these P_k^* using the Paillier private key sk. If $P_k^* = 0$, then the plaintext corresponding h_k^* to is just the final classification result. If $P_k^* \neq 0$, the user only gets the random value $r_k \cdot P_k^* + v_k$, which is the sum of v_k plus the multiplication of the random number r_k with P_k^*.

5 Security Analysis

According to the system model, the privacy preserving decision tree classification protocol should be guaranteed to be secure to the semi-honest service provider and user. In other words, the user will not disclose any information about the feature vectors and classification results during the protocols, and the service provider will not disclose any information about the parameters of the decision tree.

1. **Security for users.** The user owns the full share of the feature vector before the beginning of the protocol. During the process of step 1 in Protocol 1, the server provider owns the secret share of the bits of feature vector, but it cannot recover their value. During the secure multiplication operation via

Beaver triple, the user and the provider operate locally through their respective secret shares. By the property of the Beaver triple, it guarantees that the user and provider compute their own secret shares, i.e. the intermediate and final results of the comparison are completely hidden to each other. Thus, the user's feature data is confidential, during the calculation of the comparison results. In step 3 of the Protocol 1, the user encrypts the shares of the node comparison results via Paillier encryption scheme. As the Paillier cryptosystem is semantically secure, the eavesdropper would still not be able to obtain any information about the comparison results, even he captures the ciphertexts. Therefore, our protocol is secure for the user.

2. **Security for service provider.** The security of the provider is mainly related to Protocol 1 and Protocol 3. In Protocol 1, similar to the security analysis of user data, the secret shares of the intermediate comparison results are kept secret from each other according to the property of the Beaver triple. In step 4 of Protocol 1, as the service provider processes the share of comparison result of each decision node locally, the user does not obtain any information about the comparison results and cannot calculate the threshold value from the comparison results. In step 1 of Protocol 3, the path cost of each leaf node is randomized. The user only gets the classification result corresponding to the path cost with 0 from decryption. Thought the user can decrypt the other path costs and classification results, these values are masked by multi-plying randomness and thus the user cannot get any information about the original path costs and classification results from the decryption results. Thus, the model's thresholds and other classification data are kept confidential from the user or other external attackers.

6 Performance Evaluation

6.1 Complex Analysis

The main difference between this scheme and [10,12] is the way of handling the comparison of decision node threshold and feature value. In [10], the authors use the traditional bit-by-bit encryption, which has higher computational complexity but relatively less communication rounds. Users only need to encrypt their respective feature data bit-by-bit via a homomorphic encryption scheme, and then send them to the service provider. The service provider runs the homomorphic encryption according to the property of DGK comparison protocol. In [12], the authors use Paillier cryptosystem to encrypt data and determines the comparison results by judging between positive and negative difference values.

The comparison of the complexity and communication rounds with ours and [10,12] is shown in the following Table 1. In the table, n is the dimension of the user feature vector, m is the number of decision tree nodes of the provider, d is the depth of the decision tree, and t is the bit length of a feature value and a threshold value. Also, it is defined that the user sends the data and the provider processes the data and returns it to the user for one communication round. According to Table 1, our protocol has low computational complexity compared

with the other two schemes. While it requires too many communication rounds, the comparison algorithm of our protocol is under the modulo 2 operations and the communication complexity should not be large.

Table 1. Computational complexity.

Protocols	Complexity		Communication rounds
	User	Provider	
[10]	$O((n+m)t)$	$O((n+m)t)$	2
[12]	$O(n+m)$	$O(m)$	2
Ours	$O(m)$	$O(m)$	t

6.2 Efficiency Analysis

This protocol is evaluated on a laptop with Windows 10 operating system, 2.40GHz Intel i5-10200H processor and 16GB RAM. We used BigInteger in Java to implement our decision tree evaluation protocol, and select three datasets from the UCI database (breast cancer, heart disease and spambase) to test the computation time and communication overhead of this protocol. The evaluation results of our scheme with [10] and [12] are shown in Table 2, where n represents the number of user feature vector dimensions, d denotes the depth of the decision tree, and m denotes the number of decision nodes.

When the decision tree structure is simpler, our scheme has only a small difference in time overhead from [10] and [12]. When the decision tree is more complex, our protocol is more efficient than the other two protocols. The protocol in [10] uses Lifed ElGamal encryption scheme to encrypt data bit by bit, so the user re-quires a large amount of computation.

Table 2. Performance comparison.

Dataset	n	d	m	Protocols	Total time (s)	Bandwidth (KB)
Breast-cancer	9	8	12	[10]	1.491	64.384
				[12]	1.125	6.016
				Ours	0.642	4.768
Heart-disease	13	3	5	[10]	0.855	22.271
				[12]	0.545	3.840
				Ours	0.322	3.320
Spam-base	57	17	58	[10]	17.202	632.922
				[12]	14.911	34.560
				Ours	4.638	28.528

The protocol in [12] uses Paillier encryption scheme to encrypt data and uses the homomorphic property of Paillier encryption to compare the threshold and feature value. It reduces the computational overhead of per-bit comparison in [10], but the user and the provider still need to spend some time overhead and communication overhead to compute the ciphertext homomorphic operation and to transmit ciphertexts.

7 Conclusion

In this paper, a secure and efficient decision tree classification protocol based on an improved per-bit comparison protocol of [17]. In the decision node comparison stage, the original idea of homomorphic multiplication under ciphertext is replaced with Beaver multiplicative triples. In the process of decision tree classification, the Paillier encryption scheme is used to guarantee the privacy of user in-put, classification results and model thresholds. Through experimental analysis and comparison on real datasets, our protocol achieves low computational and communication overhead between user and service provider. We will subsequently optimize the scheme to continue reducing the computational overhead on the user side and extend the model to other machine learning algorithms such as random forests and neural networks.

Acknowledgement. This research is supported by the Basic Research Program of Qinghai Province 2020-ZJ-701.

References

1. Cock, M.D., Dowsley, R., Horst, C.: Efficient and private scoring of decision trees, support vector machines and logistic regression models based on pre-computation. IEEE Trans. Dependable Secure Comput. **16**(2), 217–230 (2019)
2. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 439–450. ACM (2000)
3. Xiong, A., Nguyen, M., So A., Chen, T.: Privacy preserving inference with convolutional neural network ensemble. In: 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC), pp. 1–6. IEEE, New York (2020)
4. Li, X., Zhu, Y., Wang, J.: On the soundness and security of privacy-preserving SVM for outsourcing data classification. IEEE Trans. Dependable Secure Comput. **15**(5), 906–912 (2018)
5. Qin, B., Li, Y., Yu, P.: Efficient privacy-preserving decision trees evaluation protocol with cloud-assisted computing. J. Xi'an Univ. Posts Telecommun. **27**(1), 1–8 (2022)
6. Hou, J., Li, Q., Meng, S., Ni, Z., Chen, Y., Liu, Y.: DPRF: a differential privacy protection random forest. IEEE Access **7**, 130707–130720 (2019)
7. Bost, R., Popa, R.A., Tu, S., et al.: Machine learning classification over encrypted data. In: Network and Distributed System Security Symposium, pp. 1–34. The Internet Society, San Diego (2015)

8. Wu, D.J., Feng, T., Naehrig, M., et al.: Privately evaluating decision trees and random forests. Proc. Priv. Enh. Technol. **2016**(4), 335–355 (2016)
9. Hsu, J.C., Tso, R., Chen, Y.C., Wu, M.E.: Oblivious transfer protocols based on commutative encryption. In: 2018 9th IFIP International Conference on New Technologies, pp. 1–5. IEEE, Paris (2018)
10. Tai, R.K.H., Ma, J.P.K., Zhao, Y., Chow, S.S.M.: Privacy-preserving decision trees evaluation via linear functions. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) ESORICS 2017. LNCS, vol. 10493, pp. 494–512. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66399-9_27
11. Zheng, Y., Duan, H., Wang, C.: Towards secure and efficient outsourcing of machine learning classification. In: Sako, K., Schneider, S., Ryan, P.Y.A. (eds.) ESORICS 2019. LNCS, vol. 11735, pp. 22–40. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29959-0_2
12. Xue, L., Liu, D.X., Huang, C., et al.: Secure and privacy-preserving decision tree classification with lower complexity. J. Commun. Inf. Netw. **5**(1), 16–25 (2020)
13. Cao, L.X., Li, Y.T., Wu, R., Guo, X., et al.: Multi key privacy protection decision tree evaluation scheme. J. Tsinghua Univ. **62**, 862–870 (2021). https://doi.org/10.16511/j.cnki.qhdxxb.2021.21.044
14. Damgard, I., Geisler, M., Krøigaard, M.: A correction to efficient and secure comparison for on-line auctions. Int. J. Adv. Comput. Technol. **1**, 323–324 (2018)
15. Garay, J., Schoenmakers, B., Villegas, J.: Practical and secure solutions for integer comparison. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 330–342. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71677-8_22
16. Beaver, D.: Efficient multiparty protocols using circuit randomization. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 420–432. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_34
17. Paillier, P.: Public-key cryptosystems based on composite degree Residuosity classes. In: Stern, J. (ed.) Advances in Cryptology — EUROCRYPT 1999. Lecture Notes in Computer Science, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
18. Sheela, M.A., Vijayalakshmi, K.: A novel privacy preserving decision tree induction. In: Information & Communication Technologies (ICT), pp. 1075–1079. IEEE, Thuckalay (2013)
19. Sumalatha, L., Sankar, P.U.: Fuzzy random decision tree (FRDT) framework for privacy preserving data mining. In: 2016 SAI Computing Conference (SAI), pp. 195–202. IEEE, London (2016)
20. Dowlin, N., Gilad-Bachrach, R., Laine, K., et al.: CryptoNets: applying neural networks to encrypted data with high throughput and accuracy. In: Proceedings of The 33rd International Conference on Machine Learning, pp. 201–210. PLMR (2016)