Efficient and Privacy-Preserving Decision Tree Classification for Health Monitoring Systems

Jinwen Liang (Graduate Student Member, IEEE), Zheng Qin (Member, IEEE), Liang Xue (Member, IEEE), Xiaodong Lin (Fellow, IEEE), and Xuemin Shen (Fellow, IEEE)

Abstract—Due to the increasing healthcare costs and the advance of wireless technology, health monitoring systems have been widely adopted recently. In health monitoring systems, a hospital outsources a clinical decision model to a cloud service provider, which receives biomedical data from remote clients and produces clinical decisions based on the outsourced model. Due to critical privacy concerns, both the clinical decision model and biomedical data should be protected. In this article, we propose an efficient and privacy-preserving decision tree (PPDT) classification scheme for health monitoring systems. Specifically, we first transform a decision tree classifier (i.e., the clinical decision model) into the Boolean vectors. Then, we leverage symmetric key encryption to encrypt the Boolean vectors as encrypted indices. The PPDT classification is achieved by searching the encrypted indices with encrypted tokens. We formulate a leakage function and provide the security definition and simulation-based proof for PPDT. The performance analyses demonstrate that PPDT is very efficient in terms of computation, communication, and storage. Experimental evaluations show that PPDT only requires microsecond-level execution time, kilobyte-level communication costs, and kilobyte-level storage costs on the test data set.

Index Terms—Cloud computing, decision tree classification, health monitoring systems, symmetric key encryption.

I. INTRODUCTION

W

ith the growing cases of chronic diseases, more and more patients have to test their health conditions constantly at hospitals, which leads to skyrocketing costs for healthcare systems [1]. To reduce the healthcare costs and improve the healthcare quality, health monitoring systems, which are often built by utilizing decision tree classification, help patients to test their health conditions periodically [2]. Coupled with recent advances of wearable devices and mobile communication networks [3], health monitoring systems work as follows: a hospital first utilizes the decision tree classification technique to produce a clinical decision model, and later tests clients’ biomedical data collected from wearable devices and provides decisions for clients based on the model [4], [5]. To further reduce the costs on the hospital side and enable practical deployment, the hospital often outsources the health monitoring services to a cloud server, which brings prominent benefits for both clients and the hospital, such as ubiquitous access, ease of management, and scalability [6].

Despite the well-known benefits, outsourcing health monitoring services to a semitrusted cloud also arises critical privacy concerns [7]–[9]. On the hospital side, since the hospital may invest a large number of resources to gather a sensitive biomedical data set and train the clinical decision model, the model is valuable intellectual property, which brings commercial benefits to the hospital. Thus, there is a demand for the hospital to protect the content of clinical decision model when outsourcing health monitoring services to the cloud service provider. On the clients’ side, both the physiological features and the clinical decision are sensitive biomedical data, because the accidental leakage of either information may reflect the clients’ health condition and lead to serious issues. For instance, if a client has a certain chronic disease, the exposure of health condition deterioration may increase the health insurance costs to the client. With the aforementioned privacy concerns, both the clinical decision model and the biomedical data should be concealed from the cloud service provider in health monitoring systems.

To protect the confidentiality of both clinical decision models and biomedical data, several privacy-preserving decision tree (PPDT) classification schemes have been proposed [10]–[20]. Most of the existing schemes are constructed based on homomorphic encryption (HE) [10]–[14] and secure multiparty computation (MPC) [15]–[19]. HE-based schemes enable PPDT classification by...
homomorphically encrypting the clinical decision model and data, which may incur prohibitive computational overheads [10]–[13]. MPC-based schemes enable multiple parties jointly and privately classify data according to decision trees, but they may lead to expensive communication costs [15]–[19]. To reduce the computation and communication overheads, Liang et al. proposed a secure decision tree classification scheme by utilizing symmetric key encryption [20]. Although the scheme in [20] achieves $O(1)$ computational complexity, it constructs huge indices, whose size is exponential to the size of the decision tree, for PPDT classification, which incurs heavy storage overheads. In summary, two main challenges should be addressed when designing PPDT classification schemes for health monitoring systems.

1) Confidentiality: Both biomedical data and clinical decision model should be protected against the cloud service provider.

2) Efficiency: The computation, communication, and storage costs should be low.

In this article, we address the aforementioned two challenges simultaneously and propose an efficient and PPDT classification scheme for health monitoring systems. First, we utilize the scheme in [21] to extract rules from decision trees by traversing all decision paths from the root node to the leaf nodes. Then, we build indices for these rules. The indices are constructed from the Boolean vectors, whose size is polynomial to the number of internal nodes, leaf nodes, and input domains. With such indices, the decision tree classification process achieves $O(1)$ computational complexity, as well as high communication and storage efficiency. After that, we propose PPDT, which incorporates symmetric key encryption, pseudorandom function, and pseudorandom permutation, to enable PPDT classification by encrypting the aforementioned indices. Accordingly, PPDT not only protects the confidentiality of both the clinical decision model and biomedical data but also achieves computation, communication, and storage efficiency for health monitoring systems. The contributions of this article are summarized as follows.

1) We propose an efficient and PPDT classification scheme for health monitoring systems. First, we transform decision tree classifiers into the Boolean vectors, which are indices that enable $O(1)$ computational complexity for decision tree classification. With such Boolean vectors, PPDT significantly improves computation, communication, and storage efficiency simultaneously. By utilizing symmetric key encryption, pseudorandom functions (Prf), and pseudorandom permutations (Prps) to protect the confidentiality of clinical decision models and biomedical data, PPDT significantly reduces the computational costs due to the adoption of low-complexity cryptographic primitives.

2) We formulate a security definition and give a simulation-based security proof for PPDT. First, we identify a leakage function $L$, which includes the size pattern, search pattern, and access pattern of PPDT. Then, we formulate the $L$-security definition, which is defined based on the leakage function $L$. Finally, we provide a simulation-based security proof to demonstrate that PPDT captures the $L$-security definition. Namely, both the clinical decision model and biomedical data are well protected.

3) We conduct performance analyses and evaluations for PPDT. We analyze the computational costs and index sizes of PPDT and the scheme in [20] (SDTC). Despite both PPDT and SDTC are with $O(1)$ computational complexity, the comparison results show that PPDT requires lower computational costs and smaller index sizes than SDTC. The experimental evaluations in the Breast-Cancer-Wisconsin data set also illustrate the performance advantages of PPDT. The performance evaluations demonstrate that: a) the computational complexity of PPDT is $O(1)$; b) PPDT only requires microseconds-level execution time, kilobyte-level communication costs, and kilobyte-level storage costs for achieving PPDT classification; and c) the performance (including computation, communication, and storage efficiency) of PPDT is orders of magnitudes boosted than SDTC.

The remainder of this article is organized as follows. Section II describes the related work. Section III provides the system model, threat model, and design goals. Section IV illustrates the preliminaries. Section V describes the construction of PPDT. Section VI formulates the leakage function and security definition, and provides a simulation-based security proof. Section VII analyzes and evaluates the performance of PPDT. Section VIII concludes this article.

II. RELATED WORK

Driven by prominent advantages, such as high accuracy, ease of deployment, and efficient evaluation, data classification techniques have been used in many application fields, such as transportation [22]–[26], malware detection [27], and healthcare [28]–[30]. In health monitoring systems, decision tree classification is often utilized to build a clinical decision model, which is further used to make decisions to biomedical features collected by wearable devices [31]. To provide services to remote clients and reduce the costs at the hospital side, health monitoring systems often require a hospital to submit the clinical decision model to a cloud services provider, and later provide health monitoring service to remote clients [19], [20]. Since a cloud service provider is not fully trusted, both the hospital and clients may worry about the privacy leakage of both the clinical decision model and biomedical data [32].

To protect the confidentiality of the clinical decision model and biomedical data in health monitoring systems, a significant amount of PPDT classification schemes have been proposed based on cryptographic tools, such as fully HE (FHE) [10], [11], additive HE (AHE) [12], [13], garbled circuit (GC) [17], [18], oblivious transfer (OT) [12], [16]–[18], secret sharing (SS) [16], [19], searchable symmetric encryption (SSE) [20], etc. We roughly divide existing PPDT classification schemes into three categories, i.e., HE-based schemes, MPC-based schemes, and SSE-based schemes.
HE-Based Schemes: Most of the HE-based schemes consider a hospital-client setting for health monitoring systems [10]–[13]. In this setting, the hospital’s clinical decision tree model is required to be protected from the client, while the client’s biomedical features and clinical predictions are required to be protected from the hospital. HE-based schemes protect the privacy of the clinical model and biomedical data by utilizing FHE [10], [11] or AHE [12], [13]. Although HE-based schemes enable PPDT classification for health monitoring systems, these schemes may face high computational costs due to high-complexity homomorphic operations [10]–[13]. To achieve efficient health monitoring services, it is desirable to design a scheme with low computational costs. Namely, achieving sublinear computational complexity and avoiding cryptographic tools with expensive computational costs.

MPC-Based Schemes: Considering a system model with multiple noncolluding hospitals or multiple noncolluding clients, most of the MPC-based schemes evaluate the prediction interactively and collaboratively. MPC-based schemes protect the privacy of both clinical decision model and biomedical data by utilizing tailored GC [15], SS [16], [19], or hybrid cryptographic tools that combine OT, SS, GC, and AHE [16]–[18]. Although MPC-based schemes avoid using high-complexity encryption, these schemes are designed based on a multiple party noncollusion security assumption and incur prohibitive communication costs due to collaboratively evaluation. To achieve real-time health monitoring services, it is important to design a novel scheme with low communication costs, i.e., avoiding MPC techniques.

SSE-Based Schemes: To improve the computational and communication efficiency for health monitoring systems, Liang et al. consider a system model that involves a hospital, a client, and a cloud service provider. In this setting, the hospital outsources the clinical decision model to a cloud service provider and, thus, the clinical decision model and biomedical data should be protected against the cloud service provider [20]. Liang et al. extracted decision rules from decision tree classifiers and developed an SSE-based scheme (SDTC) with $O(1)$ computational complexity and 1 round communication interaction [20]. Yet, since the size of indices in SDTC are exponential to the size of the decision tree classifier, SDTC suffers from prohibitive storage overheads. Furthermore, the large size of indices increase both the computational cost and the communication cost of SDTC. To achieve practical health monitoring services, it is important to design a scheme with low storage costs. Namely, the storage cost should not be exponential to the size of the input domain.

Summary: We provide Table I to summarize the differences between the aforementioned schemes. We focus on the same system setting as SDTC [20] and address all the above challenges. To achieve efficient and secure decision tree classification, we transform a decision tree classifier to indices, whose size is polynomial to the size of decision tree classifier. With such indices, the proposed SSE-based scheme is lightweight compared with SDTC in terms of computation, communication, and storage overheads.

III. MODELS AND DESIGN GOALS

A. System and Threat Models

Generally, there are three entities in health monitoring systems, i.e., a hospital ($H$), a cloud service provider ($CSP$), and a client ($C$). As shown in Fig. 1, the procedure of health monitoring systems could be described as follows.

1) Hospital ($H$): $H$ owns a clinical decision model, which is pretrained by utilizing decision tree classification techniques. By outsourcing the clinical decision model to $CSP$, $H$ offers health monitoring services to remote $C$ via $CSP$ and, therefore, could keep offline and focus on treating patients in emergency situations.

2) Client ($C$): $C$ is a patient with chronic diseases that need to be monitored periodically. With wearable sensors, $C$ regularly tests his/her biomedical features (e.g., heart rate, blood pressure, and blood oxygen concentration). Due to computation resource constraints and network connectivity challenges of wearable devices, it would not be realistic for $C$ to calculate the clinical decision for his/her biomedical features locally or to participate in the decision-making process actively by always staying online. As a result, $C$ stays offline after uploading the biomedical features to $CSP$ and later retrieves the corresponding clinical decision from $CSP$ by leveraging the cloud-empowered decision tree classification.

3) Cloud Service Provider ($CSP$): $CSP$ obtains the clinical decision model from $H$ and provides health monitoring services to $C$. When $CSP$ receives the biomedical features from $C$ periodically, $CSP$ evaluates the biomedical features and makes a decision to send back to the client.
features by utilizing the outsourced decision tree model and returns the corresponding clinical decision to \( C \).

In health monitoring systems, both \( H \) and \( C \) are honest entities, which own the clinical decision model and biomedical data (including the biomedical features and the clinical decisions), respectively. Since \( CSP \) is a third-party service provider, \( CSP \) is always viewed as a semihonest entity, which follows the procedure honestly but may be interested in the valuable clinical decision model and the sensitive biomedical data. As a result, the privacy of the clinical decision model and the biomedical data should be preserved against \( CSP \). We assume that both \( H \) and \( C \) will not collude with \( CSP \).

B. Design Goals

In this article, we aim to design an efficient and PPDT classification scheme for health monitoring systems, which should achieve the following properties.

1) Confidentiality: The confidentiality goal of PPDT is to protect sensitive data against \( CSP \), which contains two confidentiality requirements as follows.
   a) Data Confidentiality: Since biomedical features and clinical decisions are sensitive data for \( C \), the confidentiality of biomedical data should keep secret against \( CSP \).
   b) Model Confidentiality: Due to intellectual property protection issues, the clinical decision model is valuable knowledge assets for \( H \). Thus, the confidentiality of the clinical decision model should be protected against \( CSP \).

2) Efficiency: The efficiency goal of PPDT is to achieve efficient and real-time health monitoring services, which involves three requirements as follows.
   a) Low Computational Complexity: To achieve efficient and PPDT classification, the proposed scheme should achieve sublinear computational complexity and avoid using cryptographic techniques with expensive costs.
   b) Low Communication Costs: To achieve real-time health monitoring services, the communication costs of the proposed scheme should be low.
   c) Low Storage Costs: To reduce the healthcare costs of deploying the health monitoring system, the storage costs of the proposed scheme should be low.

IV. PRELIMINARIES

A. Cryptographic Preliminaries

Pseudorandom Functions: \( Prf \) are keyed functions whose outputs are computationally indistinguishable from random values [33].

Definition 1: A keyed function \( \{0,1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l \) is a \( Prf \) if for all probabilistic polynomial-time adversary \( A \), there exists a negligible function \( \text{negl} \) such that

\[
\left| \Pr_{\kappa} \left[ A^{Prf_{\kappa}}(t^1) = 1 \right] - \Pr_{\kappa} \left[ A^{\text{Rand}_{\kappa}}(t^1) = 1 \right] \right| \leq \text{negl}(t)
\]

where the key \( k \leftarrow \{0,1\}^k \) is a randomly chosen \( k \)-bit string and \( \text{Rand}_{\kappa} \) is a randomly selected from the set of functions mapping \( t \)-bit strings to \( l \)-bit strings.

Pseudorandom Permutations: \( Prp \) are keyed bijections whose output cannot be computationally distinguished from a permutation that is randomly chosen from the set of all permutations on the function’s domain [33].

Definition 2: A keyed permutation \( \{0,1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l \) is a \( Prp \) if for all probabilistic polynomial-time adversary \( A \), there exists a negligible function \( \text{negl} \) such that

\[
\left| \Pr_{\kappa} \left[ A^{Prp_{\kappa}}(t^1) = 1 \right] - \Pr_{\kappa} \left[ A^{\text{Rand}_{\kappa}}(t^1) = 1 \right] \right| \leq \text{negl}(t)
\]

where the key \( k \leftarrow \{0,1\}^k \) is a randomly chosen \( k \)-bit string and \( \text{Rand}_{\kappa} \) is selected uniformly at random from the set of all functions permuting \( t \)-bit strings to \( l \)-bit strings.

Symmetric Key Encryption: Symmetric key encryption \( \text{Sym} \) denotes any probabilistic encryption, whose encryption key is the same as the decryption key [33]. In this article, we assume \( \text{Sym} \) achieves the indistinguishability under chosen plaintext attacks (IND-CPAs).

B. Decision Tree Classification and Rule Extraction

Decision tree classification is a well-known machine learning technique used in smart healthcare applications. As shown in Fig. 2, a decision tree contains a set of internal nodes and leaf nodes, which associate with constant thresholds and predictions, respectively. Let \( t = \{t_1, t_2, \ldots, t_n\} \) be a set of thresholds associated with all internal nodes, where \( n \) denotes the number of thresholds. Let \( d = \{d_1, d_2, \ldots, d_m\} \) be a set of predictions associated with all leaf nodes, where \( m \) denote the number of predictions. With an \( n \)-dimensional feature \( f = \{f_1, f_2, \ldots, f_n\} \) as input, the decision tree classification process is described as follows. First, from the root node, the value of feature \( f_j \) is compared with the threshold \( t_j \). Second, the comparison result decides which child node (either left child node when \( f_j \leq t_j \) or right child node when \( f_j > t_j \)) should be taken for comparison next. These procedures are repeated until a leaf node \( i \), which represents the prediction is \( d_i \), reached.

Recent works, such as [20], have investigated that several decision rules could be extracted from the decision tree classifier. For example, to extract rule \( R_1 \) in Fig. 2, the decision rule extraction method proceeds as follows. First, by traversing the decision tree from the root node to the leaf node \( d_1 \), condition \( f_1 \leq t_1 \) and \( f_2 \leq t_2 \) are obtained. For the ease of description, we use a bit \( b_j = \mathbb{1}_{\{f_j > t_j\}} \) to denote the condition of comparison result of \( f_j \) and \( t_j \), where \( j \in \{1, 2, \ldots, 4\} \).
Namely, $b_1 = 0$ and $b_2 = 0$ are obtained in the first step. Second, since the conditions $b_3$ and $b_4$ are not included in the decision path from the root node to $d_1$, both $b_3$ and $b_4$ are set to $\ast$, which is a wildcard (“don’t care” value). Third, all Boolean conditions are concatenated into $b_1 \parallel b_2 \parallel b_3 \parallel b_4$, which is $00\ast\ast$ for the path from the root node to $d_1$. As a result, $R_1$ is extracted, which includes the path $00\ast\ast$ and the prediction $d_1$. By repeating the above method from the root node to each leaf node, five decision rules can be extracted from the decision tree in Fig. 2, i.e., $R = \{R_1, \ldots, R_5\}$.

With the extracted rules $R$, the process of decision tree classification can be transformed into two phases: 1) internal node comparison and 2) decision path evaluation. In the internal node comparison phase, a comparison result $c_j$ is produced after comparing each feature $f_j \in f$ with a threshold $t_j \in t$. In the decision path evaluation phase, all comparison results are concatenated as $c_1 \parallel \cdots \parallel c_n$. When the concatenation of comparison result $c_1 \parallel \cdots \parallel c_n$ matches the decision path (i.e., $b_1 \parallel \cdots \parallel b_n$) in $R_i$, the corresponding prediction for $f$ is $d_i$, where $d_i \in d$.

V. DESIGN OF PPDT

A. Definitions

Let $DT$ be a decision tree classifier with $n$ internal nodes (thresholds) and $m$ leaf nodes (predictions), which is trained from biomedical data. Let $t = \{t_1, \ldots, t_n\}$ and $d = \{d_1, \ldots, d_m\}$ be the sets of thresholds and predictions, respectively. Let $f = \{f_1, \ldots, f_n\}$ be an $n$-dimensional biomedical feature vector. Let $[w]$ be a set of integers $\{1, 2, \ldots, w\}$. Without loss of generality, we assume that all biomedical features and thresholds in decision trees are all positive integers, and the domains of them are $[w]$, i.e., $f, t \in [w]^n$. Let $R = \{R_1, R_2, \ldots, R_m\}$ be $m$ decision rules extracted from DT with $m$ leaf nodes. Each rule $R_i \in R$ is expressed as $R_i = \{R_{i,1}, \ldots, R_{i,j}, \ldots, R_{i,n}, d_i\}$, where $R_{i,j}$ denotes the value of $d_j$ in the rule $R_i$. Namely, 0, 1, and the wildcard $\ast$ are potential values for $R_{i,j}$, because $R_{i,j}$ denotes the relationship between $f_j$ and $t_j$ in the rule $R_i$. Let $V = \{v_1, \ldots, v_{1,n}, \ldots, v_{i,j}, \ldots, v_{m,1}, \ldots, v_{m,n}\}$ be $m \times n$ Boolean vectors, which represent all decision paths in $R$. Each $v_{i,j} \in V$ denotes $R_{i,j} \in R_i$ and has $w$ elements.

Let $\kappa$ be the security parameter. Let $F_0 : \{0, 1\}^k \times \{0, 1\}^{k+log w} \rightarrow \{0, 1\}^{k+log w}$ be a $prf$, where $K_{i,j} \leftarrow \{0, 1\}^k$ is the key of $F_0$. Let $H_0 : \{0, 1\}^k \times \{0, 1\}^{log(mnw)} \rightarrow \{0, 1\}^{log(mnw)}$ and $H_1 : \{0, 1\}^k \times \{0, 1\}^{log m} \rightarrow \{0, 1\}^{log m}$ be two $prf$ functions, where $K_0 \leftarrow \{0, 1\}^k$ and $K_1 \leftarrow \{0, 1\}^k$ are the keys of $H_0$ and $H_1$, respectively. Let $Sym = (Sym.Gen, Sym.Enc, Sym.Dec)$ be an IND-CPA secure symmetric key encryption, whose key space and plaintext space are $\{0, 1\}^{k+log w}$. Let $K_0$ and $K_1$ be two symmetric keys of $Sym$. Let $ER$ be a linear index for encrypted decision rules extracted from DT, which has $mnw$ elements. Let $EP$ be a linear index for encrypted predictions in DT, which has $m$ elements. $TK = \{TK_1, \ldots, TK_m\}$ are $m$ tokens for achieving the clinical decisions. Each $TK_i \in TK$ can be represented as $TK_i = (TK_{i,1}, TK_{i,2}, TK_{i,3}, L_i)$, where $L_i$ denotes $n$ locations in $ER$. All notations are summarized in Table II.

B. Boolean Vectors and Decision Trees

The basic idea of this article is to transform the DT to $m \times n$ Boolean vectors $V$ and utilize the encrypted $V$ for achieving PPDT classification. Now, we describe how to transform the decision tree classifier DT into Boolean vectors $V$ as follows.

After DT is trained from biomedical data, $\mathcal{H}$ extracts $m$ rules from DT, i.e., $R = \{R_1, \ldots, R_m\}$. For each rule $R_i \in R$, $\mathcal{H}$ builds $n$ Boolean vectors, i.e., $v_{i,1}, \ldots, v_{i,n}$. Each Boolean vector $v_{i,j} \in V$, which has $w$ elements, is used for representing $R_{i,j} \in R_i$. Then, the value of each element $v_{i,j}[k]$ in $v_{i,j}$ is set as

$$v_{i,j}[k] \leftarrow \begin{cases} 1, & \text{if } R_{i,j} = 1 \text{ and } k > t_j \\ 1, & \text{if } R_{i,j} = 0 \text{ and } k \leq t_j \\ 1, & \text{if } R_{i,j} = \ast \\ 0, & \text{otherwise.} \end{cases}$$

With the biomedical feature vector $f$ as input, the decision tree classification process in $V$ proceeds as follows. For each rule $R_i$, $\mathcal{C}$ tests $n$ values in $V$, i.e., $v_{i,1}[f_1], v_{i,2}[f_2], \ldots, v_{i,n}[f_n]$. If these values are all equal to 1, then the prediction of $f$ is $d_i$. Namely, for rule $R_i$, if $\forall v_{i,j}[f_j] = 1$, where $j \in [n]$, then the prediction of $f$ is $d_i$.

For example, the value of $R_2$ in Fig. 2 is $[0, 1, 0, \ast]$. Assume that $w = 10$ and $t = \{4, 6, 3, 7\}$, then the value of the Boolean vectors $v_{2,1}, v_{2,2}, v_{2,3}$, and $v_{2,4}$ are shown in Fig. 3. For the Boolean vector $v_{2,1}$, since $t_1 = 4$ and $R_{2,1} = 0$, elements from $v_{2,1}[1]$ to $v_{2,1}[4]$ are set to be 1, while elements from $v_{2,1}[5]$ to
to \(v_{2,1}[10]\) are set to be 0. For the Boolean vector \(v_{2,2}\), since \(t_2 = 6\) and \(R_{2,2} = 1\), elements from \(v_{2,2}[1]\) to \(v_{2,2}[6]\) are set to be 0, while elements from \(v_{2,2}[7]\) to \(v_{2,2}[10]\) are set to be 1. For the Boolean vector \(v_{2,3}\), since \(t_3 = 3\) and \(R_{2,3} = 0\), elements from \(v_{2,3}[1]\) to \(v_{2,3}[3]\) are set to be 1, while elements from \(v_{2,3}[4]\) to \(v_{2,3}[10]\) are set to be 0. For the Boolean vector \(v_{2,4}\), since \(t_4 = 7\) and \(R_{2,3} = \ast\), elements from \(v_{2,3}[1]\) to \(v_{2,3}[10]\) are set to be 1. With the feature vector \(f = \{1, 8, 3, 5\}\), since both the values of \(v_{2,1}[1], v_{2,2}[8], v_{2,3}[3]\), and \(v_{2,4}[5]\) are equal to 1, the feature vector \(f\) matches \(R_2\) and, therefore, the corresponding prediction of \(d\) is \(d_2\).

C. Detailed Design of the PPDT Scheme

The PPDT classification scheme for health monitoring systems is defined as follows.

**Definition 3 (PPDT Classification):** The PPDT involves four polynomial-time algorithms, i.e., \(\text{PPDT} = \{\text{Init}, \text{ClfEnc}, \text{TokenGen}, \text{Eva}\}\).

1. \(\text{Init}(\kappa) \rightarrow K_f, K_0, K_1, H_0, H_1, F_0, K_d\). The initialization algorithm is run by \(H\). Based on the security parameter \(\kappa\), \(H\) produces \(K_f, K_0, K_1, H_0, H_1, F_0\), and \(K_d\) and sends \(K_f, K_0, K_1, H_0, H_1, F_0\), and \(K_d\) to authorized \(C\).

2. \(\text{ClfEnc}(K_f, K_0, K_1, H_0, H_1, F_0, K_d, V, d) \rightarrow ER, ED\). The classifier encryption algorithm is run by \(H\). First, for each value in \(V\), \(H\) encrypts \(V\) and stores the encrypted values to the linear index \(ER\). Then, for each prediction \(d\), \(H\) encrypts \(d\), and stores the encrypted values to the encrypted linear index \(ED\). Finally, \(H\) outsources \(ER\) and \(ED\) to \(CSP\).

3. \(\text{TokenGen}(K_f, K_0, K_1, H_0, H_1, F_0, f) \rightarrow TK\). The token generation algorithm is run by \(C\). When \(C\) requires a clinical decision for his/her biomedical features \(f\), he/she produces a set of tokens \(TK\) for \(f\) and outsources the \(TK\) to \(CSP\).

4. \(\text{Eva}(TK, ER, ED, K_d) \rightarrow d_i\). The evaluation algorithm is an interactive algorithm run by \(CSP\) and \(C\). After receiving \(TK\) from \(C\), \(CSP\) produces a clinical decision by searching \(ER\) and \(ED\), and returns the encrypted prediction to \(C\). Then, \(C\) decrypts the encrypted prediction and receives \(d_i\) as the evaluation result for \(f_i\), where \(i = 0, 1, \ldots, t\).

**Scheme Details:** The main idea of PPDT could be described as follows. First, we construct a DT to \(m \times n\) Boolean vectors \(V\) and extract all predictions \(d\) in \(DT\), where \(d_i\) is the corresponding prediction for Boolean vectors \(v_{i,1}, \ldots, v_{i,n}\). Second, we utilize the scheme in [34] and \(\text{Sym} \) to encrypt \(V\) and \(d\), respectively. Third, we construct two indices for \(DT\) by permuting the encrypted \(V\) and \(d\) with \(\text{prP}\). Finally, PPDT classification could be achieved by searching the encrypted indices. We show the detail construction of PPDT in Fig. 4.

With PPDT, the workflow of privacy-preserving health monitoring systems contains two processes, i.e., the setup process and the decision process. In the setup process, \(H\) outsources the encrypted clinical decision model to \(CSP\) and shares several parameters to \(C\) by utilizing PPDT. First, \(H\) invokes the PPDT.Init to initialize several parameters and share these parameters to authorized \(C\). Then, \(H\) invokes the PPDT.ClfEnc to encrypt the Boolean vectors \(V\) and the medical prediction \(d\). The encrypted \(V\) and \(d\) are pseudorandomly stored in \(ER\) and \(EP\), respectively. After that, \(H\) outsources \(ER\) and \(EP\) to \(CSP\) for offering health monitoring services to remote \(C\).

In the decision process, \(C\) submits his/her biomedical features to \(CSP\) periodically and receives the clinical decision from \(CSP\). First, \(C\) invokes the PPDT.TokenGen algorithm to generate tokens \(TK\) for his/her biomedical features \(f\) and submits \(TK\) to \(CSP\). Then, \(CSP\) and \(C\) engage in the PPDT.Eva protocol. Namely, \(CSP\) returns the corresponding encrypted clinical decision by searching \(ER\) and \(EP\), while \(C\) decrypts the encrypted clinical decision and obtains the prediction for \(f\).

**Correctness:** The correctness of PPDT could be verified as follows. Assume that the prediction of \(f\) is \(d_i\), where \(i \in [m]\). We consider the following two cases for the produced token \(TK = \{TK_1, \ldots, TK_m\}\) as follows.

**Case 1:** If \(i = l\), we will have \(v_{i,j}[f_j] = 1\), where \(j \in [n]\), because \(f\) matches \(R_l\). Then, we will have \(F_0(K_f, v_{i,j}[f_j][i][i][i]|F_0(K_f, 1)[i][i][i])\). We can find that

\[
K' = \oplus_{f_j \in f} (ER[H_0[0], i][i][i]) \oplus TK_1^1
= \oplus_{f_j \in f} (F_0(K_f, v_{i,j}[f_j][i][i][i]) \oplus TK_1^1
= \oplus_{f_j \in f} (F_0(K_f, 1)[i][i][i]) \oplus TK_1^1
= K.
\]

Thus, we find that

\[
\text{Sym.Dec}(K', TK_i^2) = 0^{k+\log \kappa}
\]
and

\[
\text{Sym.Dec}(K', TK_i^1) = H_1(K_i, i).
\]

Therefore, the corresponding prediction is

\[
\text{Sym.Dec}(d_i, EP[H_1(K_i, i)]) = d_i = d_i.
\]

**Case 2:** If \(i \neq l\), we must have \(v_{i,j}[f_j] \neq 1\), for some \(j \in [n]\), because some features \(f_j \in f\) cannot satisfy \(f_j > t_j\). This, in turn, implies that for some \(j \in [n]\), \(F_0(K_f, v_{i,j}[f_j][i][i][i]|F_0(K_f, 1)[i][i][i])\). Thus, after the bitwise XOR operation, \(K' \neq K\). This ensures that with all but negligible probability, \(\text{Sym.Dec}(K', TK_i^2) \neq 0^{k+\log \kappa}\), which means \(f\) does not match \(R_l\). Thus, \(TK_i\) cannot be used for obtaining \(d_i\).

VI. Security Analysis

A. Leakage Functions and Security Definition

Before formulating the security definition of PPDT, we first define a leakage function \(L\) for PPDT, which describes
Privacy-Preserving Decision Tree Classification (PPDT)

**Initialization:** PPDT.$\text{Init}(\kappa) \rightarrow K_f, K_0, K_1, H_0, H_1, F_0, K_d$.

1. $\mathcal{H}$: $\mathcal{H}$ chooses a security parameter $\kappa$ and samples $K_f$ from $\{0, 1\}^\kappa$, i.e.,
   $$K_f \xleftarrow{\$} \{0, 1\}^\kappa.$$

   Then, $\mathcal{H}$ produces a symmetric keys $K_d$ for Sym, i.e.,
   $$K_d \leftarrow \text{Sym.Gen}(1^\kappa).$$

   After that, $\mathcal{H}$ randomly produces two pseudo-random permutation $H_0$ and $H_1$, where
   $$H_0 : \{0, 1\}^\kappa \times \{0, 1\}^{\log(m+nw)} \rightarrow \{0, 1\}^{\log(m+nw)},$$
   $$H_1 : \{0, 1\}^\kappa \times \{0, 1\}^{\log(m)} \rightarrow \{0, 1\}^{\log(m)}.$$

   $\mathcal{H}$ samples two keys $K_0$ and $K_1$ for $H_0$ and $H_1$, respectively. That is,
   $$K_0 \xleftarrow{\$} \{0, 1\}^\kappa, \quad K_1 \xleftarrow{\$} \{0, 1\}^\kappa.$$

2. $\mathcal{H} \rightarrow C$: $\mathcal{H}$ sends $K_f, K_0, K_1, K_d, H_0, H_1,$ and $F_0$ to authorized $C$.

**Classifier Encryption:** PPDT.$\text{ClfEnc}(K_f, K_0, K_1, H_0, H_1, F_0, K_d, V, d) \rightarrow ER, EP$.

1. $\mathcal{H}$: For $i \in [m], j \in [n], k \in [w]$, $\mathcal{H}$ sets:
   $$ER[H_0(K_0, i||j||k)] \leftarrow F_0(K_f, u_i,j[k])||i||j||k].$$

2. $\mathcal{H}$: For $i \in [m]$, $\mathcal{H}$ sets:
   $$EP[H_1(K_1, i)] \leftarrow \text{Sym.Enc}(K_d, d_i),$$

   where $d_i \in d$.

3. $\mathcal{H} \rightarrow CSP$: $\mathcal{H}$ outsources $ER$ and $EP$ to $CSP$.

**Token Generation:** PPDT.$\text{TokenGen}(K_f, K_0, H_0, F_0, f) \rightarrow TK$.

1. $C$: When $C$ requests a clinical decision for his/her biomedical features $f = \{f_1, f_2, \ldots, f_n\}$, he/she samples a key $K$ for Sym, i.e.,
   $$K \xleftarrow{\$} \{0, 1\}^\kappa.$$

   Then, he/she produces $m$ tokens $TK = \{TK_1, TK_2, \ldots, TK_m\}$. Token $TK_i$ contains three values and a vector, i.e., $TK_i = (TK_i^1, TK_i^2, TK_i^3, L_i)$, which are calculated as follows.
   $$TK_i^1 = \oplus_{j \in f}(F_0(K_f, 1||i||j||f_j) \oplus K),$$
   $$TK_i^2 = \text{Sym.Enc}(K, 0^\kappa + \log \kappa),$$
   $$TK_i^3 = \text{Sym.Enc}(K, H_1(K_1, i)),$$
   $$L_i = \{H_0(K_0, i||j||f_j)\}_{j \in [n]}.$$

2. $C \rightarrow CSP$: $C$ submits $TK$ to $CSP$.

**Evaluation:** PPDT.$\text{Eval}(TK, ER, ED, K_d) \rightarrow d_i$.

1. $CSP$: $CSP$ receives $ER$ and $EP$ from $\mathcal{H}$.

2. $CSP$: After receiving $TK$ from $C$, for $i \in [m]$, $CSP$ calculates
   $$K' = \oplus_{j \in f}(ER[H_0(K_0, i||j||f_j)]) \oplus TK_i^1.$$

3. $CSP \rightarrow U$: If $\text{Sym.Dec}(K', TK_i^2) = 0^\kappa + \log \kappa$, then $CSP$ searches $EP[\text{Sym.Dec}(K', TK_i^2)]$ and obtains $\text{Sym.Enc}(K_d, d_i)$, where $i \in [m]$. Then, $CSP$ returns $\text{Sym.Enc}(K_d, d_i)$ to $U$.

4. $U$: After receiving $\text{Sym.Enc}(K_d, d_i)$, $U$ calculates
   $$d_i = \text{Sym.Dec}(K_d, \text{Sym.Enc}(K_d, d_i)),$$

   where $i \in [m]$. Finally, $d_i$ is the corresponding clinical decision for $f$.

---

The leaked information in $L(V, f)$ is usually considered default leaked information in most of the searchable encryption schemes and secure decision tree classification scheme. With the leakage function $L(V, f)$, the adaptive $L$-security definition can be formulated.

**Definition 5:** Adaptive $L$-security. Let $\Phi = (\text{Init}, \text{ClfEnc}, \text{TokenGen}, \text{Eval})$ be a PPDT classification scheme. Let $A = (A_0, A_1, \ldots, A_q)$ and $S = (S_0, S_1, \ldots, S_q)$ be an adversary and a simulator, respectively, where $q \in \mathbb{N}$. Let $f^1, f^2, \ldots, f^q$ be biomedical feature vectors for $q$ PPDT classification requests generated by $A$. We define the $\text{Real}^A_{L, S}(1^\kappa)$ experiment and the $\text{Sim}^A_{L, S}(1^\kappa)$ experiment as follows.

**Real}^A_{L, S}(1^\kappa):** At round 0, the challenger invokes $\text{Init}(\kappa)$ to produce $K_f, K_0, K_1, H_0, H_1, F_0,$ and $K_d$. Then, $A_0$ produces a decision tree $DT$ and extracts Boolean
vectors $V$ and predictions $d$ from DT. The challenger invokes $\text{ClfEnc}(K_f, K_0, K_1, H_0, H_1, F_0, K_d, V, d)$ to generate encrypted indices $ER$ and $EP$, and sends $ER$ and $EP$ to $A$. After that, $A$ makes $q$ classification requests. At round $r$ ($1 \leq r \leq q$): $A_r$ reviews the previous requests and generates $f^*$ adaptively. The challenger invokes $\text{TokenGen}(K_f, K_0, H_0, H_1, F_0, F_0, f) \rightarrow \text{TK}^r$, and submits $\text{TK}^r$ to $A_r$. Then, $A_r$ searches the encrypted indices $ER$ and $EP$ by utilizing $\text{TK}^r$ and obtains an encrypted prediction for $f^*$. After $q$ round interactions, $A$ produces a bit as the output.

$Sim^A_{L,S}(1^*)$: At round $0$, $S_0$ randomly generates two indices $ER^*$ and $EP^*$ by utilizing $L(V, f)$ and sends both $ER^*$ and $EP^*$ to $A$. Then, $A$ makes $q$ classification requests. At round $r$ ($1 \leq r \leq q$): $A_r$ reviews the previous requests and generates $f^*$ adaptively. With $L(V, f)$, $S_r$ generates $m$ appropriate tokens $\text{TK}^{rs} = \{\text{TK}^{r_1s}, \ldots, \text{TK}^{r_ms}\}$, where $\text{TK}^{rs} = (\text{TK}^{r_1}, \text{TK}^{r_2}, \text{TK}^{r_3}, \text{L}_0)$. After that, $S_r$ submits $\text{TK}^{rs}$ to $A_r$. Then, $A_r$ searches the encrypted indices $ER^*$ and $EP^*$ by utilizing $\text{TK}^{rs}$, and obtains an encrypted prediction for $f^*$. After $q$ round interactions, $A$ produces a bit as the output.

We say that $\Phi$ is adaptively $\mathcal{L}$-secure if for all polynomial size adversaries $A = (A_0, \ldots, A_q)$, there exist a simulator $S = (S_0, \ldots, S_q)$ and a negligible function $\text{negl}(\kappa)$ such that

$$\text{Pr}[\text{Real}^i_{S_0}(1^*) = 1] - \text{Pr}[\text{Sim}^A_{L,S}(1^*) = 1] \leq \text{negl}(\kappa).$$

B. Security Proofs

**Theorem 1:** $\text{PPDT}$ is adaptively $\mathcal{L}$-secure if $H_0$ and $H_1$ are $\text{Prps}$, $F_0$ is a $\text{Prf}$, and $\text{Sym}$ is a IND-CPA secure symmetric key encryption.

**Proof:** We create a simulator $S = (S_0, \ldots, S_q)$ such that for an adversary $A = (A_0, \ldots, A_q)$, the outputs of $\text{Real}^i_{S_0}(1^*)$ and $\text{Sim}^A_{L,S}(1^*)$ are computationally indistinguishable. The simulator $S = (S_0, \ldots, S_q)$ that adaptively produces $ER^*$, $EP^*$, and $\text{TK}^{rs}$ is constructed as follows.

$S_0$: With the size pattern in leakage function $L(V, f)$, $S_0$ obtains $m$, $n$, and $w$. To produce $ER^*$, $S_0$ constructs a linear index with $m$ elements. For each element in $ER^*$, $S_0$ sets the value as a random string $\{0, 1\}^{k + \log x}$. Namely, each random value in $ER^*$ has the same length as the output of $F_0$. To produce $EP^*$, $S_0$ constructs a linear index with $m$ elements. For each element in $EP^*$, $S_0$ sets the value as a random string $\{0, 1\}^{k + \log x}$. Namely, each random value in $EP^*$ has the same length as the output of $\text{Sym.Enc}$. Finally, $S_0$ sends both $ER^*$ and $EP^*$ to $A_0$. With all but negligible probability, $A$ cannot obtain $K_f$ and, thus, $A$ cannot distinguish the pseudorandom output of $F_0(K_f, 1||j||k)$ in $ER$ from a $\{0, 1\}^{k + \log x}$ bit random string in $ER^*$, where $i \in [m]$, $j \in [n]$, and $k \in [w]$. Meanwhile, with all but negligible probability, $A$ cannot obtain $K_d$ and, therefore, $A$ cannot distinguish the value of $\text{Sym.Enc}(K_d, d^i)$ in $EP$ from a $\{0, 1\}^{k + \log x}$ bit random string in $ER^*$, if $\text{Sym}$ is an IND-CPA secure symmetric encryption. Hence, both $ER^*$ and $EP^*$ are computationally indistinguishable from $ER$ and $EP$, respectively.

$S_r$: For $1 \leq r \leq q$: With the search pattern of the input biomedical feature vector in $L(V, f)$, $S_r$ checks whether the biomedical feature vector $f^*$ has appeared before. There are two different cases.

a) The biomedical feature vector $f^*$ has totally appeared before. Namely, there exists $f^*$, such that $f^* = f^* \in [n]$, $1 \leq t < r$. Then, $S_r$ searches the access patterns in leakage function $L(V, f)$, and returns $\text{TK}^{rs} = \text{TK}^{rs}$ as the appropriate token to $A_r$ for $f^*$.

b) Some values (or all values) of the biomedical feature vector $f'$ have not appeared before. The token $\text{TK}$ could be generated as follows. First, $S_r$ randomly chooses an element $u$ in $EP^*$. Second, token $\text{TK}^{rs} = (\text{TK}^1, \text{TK}^2, \text{TK}^3, \text{L}_0)$ is generated as follows.

i) With the search pattern in $L(V, d)$, $S_r$ checks whether $f^*$ has appeared before. If $f^* \in f'$ has appeared before, it means that there exists $f^*_i = f^*_i$, where $1 \leq t < r$. $S_r$ selects the same location of $f^*_i$ for token $u$ in $ER^*$ as $L^{rs}_u$. Otherwise, $S_r$ randomly selects a location that have not been selected in $ER^*$ as $L^{rs}_u$. Finally, $S_r$ produces $L^{rs}_u = \{L^{rs}_{u1}, \ldots, L^{rs}_{u1}, \ldots, L^{rs}_{un}\}$.

ii) $S_r$ randomly samples a $\text{Sym}$ key $K$. Then, with $L^{rs}_u = \{L^{rs}_{u1}, \ldots, L^{rs}_{u1}, \ldots, L^{rs}_{un}\}$, $S_r$ calculates

$$\text{TK}^{rs} = \oplus_{L^{rs}_u \in L^{rs}_u} \text{Enc}(\text{ER}^{\text{sym}}(L^{rs}_u)) \oplus K^\ast.$$
A. Performance Analysis

The performance of PPDT depends on several parameters. We provide a list of parameters that is needed for performance analysis in Table III. Note that once a decision tree DT is trained, $m$, $n$, and $w$ are constants.

In Table IV, we summarize and compare the computational performance properties of PPDT with the privacy-preserving decision tree classification scheme in [20] (SDTC) in terms of the computational costs of algorithms, including Init, ClfEnc, TokenGen, and Eva, the size of both indices and tokens.

### Table III: Parameters for Performance Analysis

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_{\text{m}}$</td>
<td>The number of leaf nodes of DT.</td>
</tr>
<tr>
<td>$n_{\text{w}}$</td>
<td>The number of internal nodes of DT.</td>
</tr>
<tr>
<td>$w_{\text{m}}$</td>
<td>The domain of biometric features.</td>
</tr>
<tr>
<td>$T_{\text{gen}}$</td>
<td>Computational cost of generating a Sym key.</td>
</tr>
<tr>
<td>$T_{\text{enc}}$</td>
<td>Computational cost of Sym encryption.</td>
</tr>
<tr>
<td>$T_{\text{dec}}$</td>
<td>Computational cost of Sym decryption.</td>
</tr>
<tr>
<td>$T_{\text{enc}}^{\text{prf}}$</td>
<td>Computational cost of generating a prf key.</td>
</tr>
<tr>
<td>$T_{\text{enc}}^{\text{prf}}$</td>
<td>Computational cost of calculating a prf.</td>
</tr>
<tr>
<td>$T_{\text{enc}}^{\text{prf}}$</td>
<td>Computational cost of generating a prp key.</td>
</tr>
<tr>
<td>$T_{\text{enc}}^{\text{prp}}$</td>
<td>Computational cost of calculating a prp.</td>
</tr>
<tr>
<td>$T_{\text{enc}}^{\text{prf}}$</td>
<td>Computational cost of performing an exclusive-or operation.</td>
</tr>
<tr>
<td>$S_{\text{sym}}$</td>
<td>Size of Sym ciphertexts.</td>
</tr>
<tr>
<td>$S_{\text{prf}}$</td>
<td>Size of prf outputs.</td>
</tr>
<tr>
<td>$S_{\text{prp}}$</td>
<td>Size of prp outputs.</td>
</tr>
</tbody>
</table>

### Computation Costs: Since all parameters in Table IV are constants, the computation complexity of both PPDT and the scheme in [20] (SDTC) is $O(1)$. As shown in Table IV, the computational costs of the initialization algorithm of both PPDT and SDTC are the same. Meanwhile, Table IV demonstrates that the computational costs of the classifier encryption algorithm of SDTC is $(w^3) \cdot (2 \cdot T_{\text{prf}} + T_{\text{prp}} + T_{\text{enc}} + T_{\text{xor}})$, which is an exponential computational cost. By constructing different indices for the decision tree classifier, the computational cost of the classifier encryption algorithm of PPDT is $m \cdot n \cdot w \cdot (T_{\text{prf}} + T_{\text{prp}}) + m \cdot (T_{\text{enc}} + T_{\text{prp}})$, which is significantly boosted to polynomial costs. Also, the computational overheads of the token generation algorithm and evaluation algorithm of PPDT are $T_{\text{gen}} + m \cdot ((n+1)T_{\text{xor}} + n \cdot T_{\text{prf}} + T_{\text{prp}}) + 2 \cdot T_{\text{enc}}$ and $m \cdot (n+1) \cdot T_{\text{xor}} + (O(n)+2) \cdot T_{\text{dec}}$, which can provide efficient computation by utilizing low-complexity Sym, prf, and prp. Therefore, PPDT not only significantly improves the computational cost of classifier encryption from exponential to polynomial but also achieves efficient computation and makes the health monitoring services more practical.

### Storage and Communication Costs: We also provide the theoretical storage and communication costs in Table IV by analyzing the size of indices and tokens. Since the encrypted indices are required to submit by $\mathcal{H}$ and stored at $\mathcal{CSP}$, the size of indices denotes both the communication costs of $\mathcal{H}$ and the storage costs of $\mathcal{CSP}$. SDTC requires $2 \cdot (w^3) \cdot (S_{\text{sym}} + S_{\text{prf}})$ to store the indices, and PPDT only requires $m \cdot n \cdot w \cdot (S_{\text{prf}} + S_{\text{prp}}) + m \cdot S_{\text{sym}}$ to store the tokens. Thus, PPDT boosts the exponential storage efficiency to polynomial storage efficiency. The size of tokens denotes the communication costs of $\mathcal{C}$, which sends encrypted biomedical features to $\mathcal{CSP}$. Table IV shows that the size of tokens of PPDT is $m \cdot (S_{\text{prf}} + 2 \cdot S_{\text{sym}} + n \cdot S_{\text{prp}})$, which is also efficient for health monitoring systems. Thus, the PPDT achieves storage and communication efficiency.

B. Experimental Settings

We conduct experimental evaluations in the real data set to show the performance advantage of PPDT. PPDT is implemented in C++ code based on OpenSSL.\(^1\) The experiments are conducted on a 64-bit VMware Workstation (running Ubuntu 18.04) with an Intel Core i7-8850H CPU with 2.60-GHz and 8-GB RAM. AES-CBC-256 and HMAC-256 are utilized to implement Sym and prf, respectively. The prp is implemented by utilizing prf.

### Data Sets: We utilize the Breast-Cancer-Wisconsin\(^2\) data set to train the clinical decision model for health monitoring systems. The Breast-Cancer-Wisconsin data set includes 683 nonmissing biomedical data records. Each record has nine discrete attributes, whose values are located in the integer set $\{1, \ldots, 10\}$.

### Decision Trees. The decision trees are trained in plaintext form by using scikit-learn.\(^3\) The CART decision tree classification technique is utilized for training five decision

\(^{1}\)https://www.openssl.org/  
\(^{3}\)https://scikit-learn.org
trees from the Breast-Cancer-Wisconsin data set. The numbers of leaf nodes \( m \), decision nodes \( n \), and plaintext domain \( w \) are listed as follows: 1) \( \text{DT}_1: m = 4, n = 3, \) and \( w = 10; \) 2) \( \text{DT}_2: m = 5, n = 4, \) and \( w = 10; \) 3) \( \text{DT}_3: m = 7, n = 6, \) and \( w = 10; \) 4) \( \text{DT}_4: m = 10, n = 9, \) and \( w = 10; \) and 5) \( \text{DT}_5: m = 12, n = 11, \) and \( w = 10. \) We provide Table V to show the true positive (TP), false positive (FP), false negative (FN), true negative (TN), and decision accuracy of the aforementioned decision trees. As shown in Table V, although parameters, such as \( m, n, \) and \( w \) are small, the accuracy of each model is high (achieves an accuracy that more than 95%). In real-world applications, if these parameters are larger, the clinical decision model may become overfitting. Thus, we utilize the above five clinical decision model to evaluate the performance advantages of PPDT.

**Baselines:** We compare the performance advantages of PPDT with the scheme in [20] (SDTC). Similar to PPDT, SDTC also achieves \( O(1) \) computational complexity and is designed based on symmetric key encryption, Prfs, and Prps. The main difference between PPDT and SDTC is the construction of indices. As shown in Table IV, the size of indices in PPDT is polynomial to \( m, n, \) and \( w, \) while the size of indices in SDTC is exponential to \( w. \) Although \( m, n, \) and \( w \) are constants, the computational, communication, and storage overheads of PPDT and SDTC are different. In the comparison experiments, SDTC is also implemented in the same setting as PPDT.

**C. Experimental Evaluations**

We make 500 experiment runs to evaluate the total time cost of each algorithm of PPDT, which are shown in Fig. 5. For each algorithm, we test \( \text{DT}_1, \text{DT}_2, \text{DT}_3, \text{DT}_4, \) and \( \text{DT}_5 \) that are trained from the Breast-Cancer-Wisconsin data set. Fig. 5(a) illustrates that the total time cost of the initiation algorithm grows linearly when the number of systems grows linearly, which demonstrates that the time complexity of the \( \text{Init} \) algorithm of PPDT is \( O(1). \) Fig. 5(b) shows that the total time cost of the classifier encryption algorithm grows linearly when the number of decision tree classifiers grows linearly, which illustrates that the time complexity of the \( \text{ClfEnc} \) algorithm of PPDT is \( O(1). \) Fig. 5(c) demonstrates that the total time cost of the token generation algorithm grows linearly when the number of biomedical feature vectors grows linearly, which describes that the time complexity of the \( \text{TokenGen} \) algorithm of PPDT is \( O(1). \) Fig. 5(d) demonstrates that the total time cost of the evaluation algorithm grows linearly when the number of decision requests grows linearly, which shows that the time complexity of the \( \text{Eva} \) algorithm of PPDT is \( O(1). \) Therefore, Fig. 5 shows that the time complexity of each algorithm in PPDT is \( O(1), \) which demonstrates that PPDT achieves the faster-than-linear time complexity.

We compare PPDT with SDTC, which enables \( O(1) \) computational complexity for secure decision tree classification. We evaluate performance differences between PPDT and SDTC by testing the average time costs at hospital (H), cloud service provider (CSP), and clients (C), the communication costs at H and C, and the storage costs at CSP. Table VI shows the performance advantages of PPDT compared with SDTC on the Breast-Cancer-Wisconsin data set. Note that due to the heavy storage and communication overhead, we estimate the performance of SDTC in case \( \text{DT}_4 \) and \( \text{DT}_5 \) by utilizing the results in Table IV and the tested results of parameters in Table III. Compared with SDTC, Table VI shows that PPDT boosts the efficiency in terms of average time costs. With polynomial

**TABLE IV**

<table>
<thead>
<tr>
<th>Performance Properties</th>
<th>PPDT</th>
<th>The Scheme in [20] (SDTC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost of Initialization</td>
<td>( T_{\text{Init}} + T_{\text{ClfEnc}} + 2 \cdot T_{\text{TokenGen}} )</td>
<td>( T_{\text{Init}} + T_{\text{ClfEnc}} + 2 \cdot T_{\text{TokenGen}} )</td>
</tr>
<tr>
<td>Cost of Classifier Encryption</td>
<td>( m \cdot n \cdot w \cdot (T_{\text{Prep}} + T_{\text{ClfEnc}}) + m \cdot (T_{\text{TokenGen}} + T_{\text{Prep}}) )</td>
<td>( (w^m) \cdot (2 \cdot T_{\text{Prep}} + T_{\text{ClfEnc}} + T_{\text{TokenGen}} + T_{\text{Prep}}) )</td>
</tr>
<tr>
<td>Cost of Token Generation</td>
<td>( T_{\text{TokenGen}} + m \cdot ((n + 1) T_{\text{Prep}} + n \cdot (T_{\text{Prep}} + T_{\text{ClfEnc}}) + 2 \cdot T_{\text{TokenGen}}) )</td>
<td>( T_{\text{TokenGen}} + 2 \cdot (T_{\text{Prep}} + T_{\text{ClfEnc}}) )</td>
</tr>
<tr>
<td>Size of Indexes</td>
<td>( m \cdot n \cdot w \cdot (S_{\text{Prep}} + m \cdot S_{\text{Prep}}) )</td>
<td>( 2 \cdot (w^m) \cdot (S_{\text{Prep}} + S_{\text{Prep}}) )</td>
</tr>
<tr>
<td>Size of Tokens</td>
<td>( m \cdot (S_{\text{Prep}} + 2 \cdot S_{\text{Prep}} + n \cdot S_{\text{Prep}}) )</td>
<td>( S_{\text{Prep}} + S_{\text{Prep}} )</td>
</tr>
</tbody>
</table>

**TABLE V**

<table>
<thead>
<tr>
<th>Models</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>TN</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{DT}_1 )</td>
<td>423</td>
<td>10</td>
<td>21</td>
<td>229</td>
<td>95.46%</td>
</tr>
<tr>
<td>( \text{DT}_2 )</td>
<td>423</td>
<td>6</td>
<td>21</td>
<td>233</td>
<td>96.05%</td>
</tr>
<tr>
<td>( \text{DT}_3 )</td>
<td>422</td>
<td>3</td>
<td>22</td>
<td>236</td>
<td>96.34%</td>
</tr>
<tr>
<td>( \text{DT}_4 )</td>
<td>433</td>
<td>6</td>
<td>11</td>
<td>233</td>
<td>97.51%</td>
</tr>
<tr>
<td>( \text{DT}_5 )</td>
<td>434</td>
<td>5</td>
<td>10</td>
<td>234</td>
<td>97.80%</td>
</tr>
</tbody>
</table>

*Fig. 5.* Time costs of PPDT. (a) Time cost of initiation. (b) Time cost of classifier encryption. (c) Time cost of token generation. (d) Time cost of evaluation.
size indices, the computational efficiency of PPDT is orders of magnitude faster than SDTC, which utilizes indices with exponential size, in terms of total time costs. Specifically, by reducing the size of indices, PPDT improves the computational efficiency of classifier encryption algorithm and, thus, significantly reduces the time cost at the $H$ side. Hence, PPDT only requires several microseconds for $H$, CSP, and C to finish privacy-preserving health monitoring processes with decision tree classification. Meanwhile, Table VI demonstrates that the communication costs of PPDT is about several kilobytes, which is practical for health monitoring services. For DT$_1$ in Table VI, which has 11 internal nodes and 12 leaf nodes, PPDT only requires 42.6-KB communication cost at $H$ side and 1.2 KB at $C$ side, but SDTC requires 12.8 TB at $H$ side and 33.4 Bytes at $C$ side. Thus, it is clear that PPDT is communication-efficient for health monitoring services. Finally, with small size indices for encrypted decision trees, PPDT significantly reduces the storage cost at the CSP side, which benefits health monitoring services. Therefore, PPDT is a computational, communication, and storage efficient scheme for privacy-preserving health monitoring systems.

### VIII. Conclusion

In this article, we have proposed PPDT, which enables privacy-preserving decision tree classification for health monitoring systems. Different from the existing schemes, PPDT transforms a decision tree classifier into the Boolean vectors and utilizes symmetric encryption to protect the data privacy. With such a design, PPDT extremely boosts the computational, communication, and storage efficiency. We have formulated a leakage function $L$, provided the adaptively $L$-security definition, and given a simulation-based security proof for PPDT. Performance analyses demonstrated that PPDT achieves $O(1)$ computational complexity with polynomial-size indices. Experimental evaluations demonstrated that PPDT achieves microsecond-level execution time, kilobyte-level communication costs, and kilobyte-level storage costs on the Breast-Cancer-Wisconsin data set. Consequently, we have addressed both the confidentiality and efficiency challenges simultaneously, and PPDT is an efficient, practical, and real-time solution for health monitoring systems. For the future works, we will design a PPDT classification scheme against malicious adversaries.

### References


Liang Xue received the B.S. and M.S. degrees from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2015 and 2018, respectively. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

Her research interests include applied cryptography, cloud computing, and blockchain.

Xiaodong Lin (Fellow, IEEE) received the Ph.D. degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1998, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2008.

He is currently an tenured Associate Professor with the School of Computer Science, University of Guelph, Guelph, ON, Canada. His research interests include wireless network security, applied cryptography, computer forensics, and software security.

Zhenqin (Member, IEEE) received the Ph.D. degree in computer science from Chongqing University, Chongqing, China, in 2001. He was a Visiting Scholar with Michigan State University, East Lansing, MI, USA, from 2010 to 2011. He is a Full Professor and the Vice Dean with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. He is the Director of the Hunan Key Laboratory of Big Data Research and Application, and the Vice Director of the Hunan Engineering Laboratory of Security, Changsha. His research interests include information security, and software engineering.

Jinwen Liang (Graduate Student Member, IEEE) received the B.S. degree in information security from Hunan University, Changsha, China, in 2015, where he is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering.

He is also a visiting Ph.D. student with BBCR Laboratory, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include applied cryptography, blockchain, order-preserving encryption, and secure data classification.

Mr. Liang served as the TPC Member for IEEE VTC 2019 Fall.