IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 14, 2022, accepted January 31, 2022, date of publication February 7, 2022, date of current version February 16, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3149771

Private and Energy-Efficient Decision Tree-Based Disease Detection for Resource-Constrained Medical Users in Mobile Healthcare Network

SONA ALEX, (Member, IEEE), K. J. DHANARAJ, (Member, IEEE), AND P. P. DEEPTHI, (Senior Member, IEEE)
Department of Electronics and Communication Engineering, National Institute of Technology Calicut, Kozhikode 673601, India
Corresponding author: Sona Alex (sona_p160088ec@nitc.ac.in)

ABSTRACT In mobile healthcare networks (MHN), outsourced disease detection services demand the privacy preservation of medical users and health service providers (health clouds). This necessitates the use of a fully homomorphic encryption (FHE) while providing disease detection services, such as decision tree-based disease detection. However, the existing homomorphic encryption schemes utilized in decision tree-based disease detection that ensure the privacy of the medical user and health cloud are computationally-intensive and energy-hungry at the edge devices. Hence the medical user finds it difficult to exploit the existing private decision tree-based disease detection services due to restrictions on battery capacity and computing resources. Therefore, this work proposes a protocol for private decision tree classification with low resource consumption (PDTC-LRC) on edge devices of medical users by considering decision tree parameters as confidential to the health cloud. An energy-efficient, additively homomorphic, symmetric key-based FHE-compatible Rivest scheme (FCRS) is developed for implementing PDTC-LRC. FCRS can be decrypted homomorphically at the health cloud to support additive and multiplicative homomorphism. Also, an energy and bandwidth-efficient secure integer comparison protocol is developed for realizing PDTC-LRC. Experiments on the Raspberry Pi 3B+ board validate the improved energy efficiency and real-time applicability of the proposed secure integer comparison protocol and decision tree classifier compared with similar schemes available in the literature. Simulation and mathematical analysis ensure that user and health cloud privacy requirements are achieved by maintaining the classification accuracy same as that of decision tree classification in the plain domain.

INDEX TERMS Mobile healthcare networks, privacy, decision tree algorithm, homomorphic encryption.

I. INTRODUCTION

Mobile healthcare network (MHN) consists of wearable devices, medical users, cloud servers, and heterogeneous mobile networks. MHN improves healthcare quality by continuously monitoring personal health information (PHI) such as heart rate, respiration rate, and blood pressure [1]. In MHN, cloud servers facilitate computer-aided remote disease predictions by training machine learning (ML) models and making predictions based on the PHI received from the medical user. However, concern about the loss of privacy and security of PHI is a barrier to the adoption of cloud services by medical users [2], [3]. Moreover, the cloud keeps disease detection algorithms confidential due to competitive advantages. Hence, ML-based detection should be performed in the secure domain to ensure the data privacy of the medical user and cloud servers in MHN.

The user devices in MHN face severe restrictions on battery power and computing resources [4], [5]. However, private ML-based disease diagnoses using decision tree (DT) algorithms have been established [6]–[9], using energy-hungry homomorphic encryption (HE) schemes. As DT algorithms require both addition and multiplication operations, it may become essential to use fully homomorphic encryption (FHE) schemes to preserve the privacy of both the medical user and cloud [6], [7]. Due to their complexity, the public key FHE schemes are not suitable for resource-constrained users in MHN. Also, as the number of multiplications required to realize DT classifier (DTC) increases, computational complexity and ciphertext expansion of the FHE scheme increase. To achieve resource utilization efficiency at the user side, the encryption at the user edge

The associate editor coordinating the review of this manuscript and approving it for publication was Larbi Boubchir.

device can be performed with a simple FHE-compatible cipher [10], [11]. An FHE-compatible cipher is an energy and bandwidth-efficient symmetric key cipher. The data encrypted with an FHE-compatible cipher can be homomorphically decrypted to FHE encrypted form. The medical user can send data encrypted using an FHE-compatible scheme to the cloud. At the cloud, homomorphic decryption converts the received encrypted data to FHE encrypted form for private DTC operation.

The existing FHE-compatible ciphers [10], [11] do not support the integer arithmetic on encrypted data which is essential for private DTC operations. Also, homomorphic decryption of these ciphers is time-consuming. Alternative for encryption at resource constrained edge device is symmetric key FHE schemes [12], [19]. However, symmetric key FHE schemes available in literature can't ensure the privacy of both medical user and health cloud while performing privacy-preserving ML-based processing. Therefore, for improved resource efficiency at the user side, it is required to deploy real-time, low-power, and secure ciphers at the edge device. Also, these low-power ciphers should support homomorphic addition and multiplication required for disease detection at cloud. However, using homomorphic properties of FHE, only linear operations that involve addition and multiplication can be realized in the encrypted domain. Secure multiparty computation (SMC) can be performed with the homomorphically encrypted data in order to carry out nonlinear operations such as integer comparison in the encrypted domain. Moreover, the outsourced disease diagnosis procedure needs to be redesigned based on these low-power encryption schemes and SMC for improved resource efficiency at the user side.

In this work, we address the challenge of preserving the privacy and accuracy of the decision tree-based disease detection with very low computational and battery power requirements for the MHN user device at acceptable levels of delay. The main contributions of this paper can be summarized as follows,

- Energy and bandwidth-efficient, additively homomorphic, symmetric key-based FHE-compatible Rivest scheme (FCRS) is proposed for data encryption at the user side. FCRS is developed by modifying Rivest *et al.'s* encryption scheme [12]. The data encrypted with the proposed FCRS can be decrypted homomorphically in such a way as to support both the encrypted domain integer multiplication and addition required for secure disease diagnosis at the cloud.
- A SMC protocol for private decision tree classification with low resource consumption (PDTC-LRC) is proposed based on FCRS for secure domain disease detection such that the privacy of both the medical user and the health cloud are protected. For PDTC-LRC, a novel SMC-based secure integer comparison protocol is developed which improves resource efficiency by reducing the number of handshaking operations between the user and health cloud.
- Security analysis is performed to demonstrate that the proposed FCRS can resist possible attacks (ciphertext only attack and known-plaintext attack) and achieve semantic security. Also, it is validated that while transferring FCRS secret keys to the health cloud, man-in-the-middle attacks, and spoofing attacks can be resisted. The formal privacy analysis is performed using a simulation model (real vs. ideal) to show that proposed SMCs preserve the data privacy of the user and health cloud.

The remnant of this paper is arranged as follows: An outline of the related work is given in Section II. The preliminaries of the proposed protocol are furnished in Section III. The system specifications are presented in Section IV. The proposed symmetric key encryption scheme with low resource consumption is described in Section V. Section VI describes the proposed private decision tree-based classification developed based on the proposed symmetric key encryption scheme and secure integer comparison protocols. The security analysis of the proposed symmetric key encryption scheme as well as the formal privacy analysis of the proposed protocols are presented in Section VII. The efficiency analysis of the proposed algorithms through extensive simulations and implementations on the Raspberry Pi 3B+ board is detailed in Section VIII. Section IX concludes the paper.

II. RELATED WORKS

The partially or fully homomorphic public-key cryptographic constructions [6], [8], [9], [14]–[16] which are used for private disease detection algorithms are energy-hungry, and therefore are not suitable for resource-constrained user devices. The symmetric key encryption schemes such as secret sharing scheme (SSS) [17], Rivest *et al.'s* symmetric encryption scheme (RSE) [12], modified Rivest *et al.'s* symmetric encryption scheme (MRSE) [13], symmetric homomorphic encryption (SHE) [18] and single key fully homomorphic data encapsulation mechanism (SFH-DEM) [19] are suitable for resource-constrained user devices. SFH-DEM supports additive and multiplicative homomorphism. However, SFH-DEM cannot preserve the privacy requirements of the health cloud while performing private ML algorithms. The encryption operation of SFH-DEM requires the knowledge of functions to be performed using the homomorphism of SFH-DEM. The RSE supports additive and multiplicative homomorphism [12]. However, the same property weakens the scheme through vulnerability to ciphertext-only attacks. MRSE, SHE, and SSS schemes support only additive homomorphism. However, outsourced decision tree-based disease diagnosis demands additive and multiplicative homomorphism. The existing FHE schemes ([6], [14], [15]) that support security requirements of the medical user and preserve the privacy of the health cloud while performing disease diagnosis are not suited for use in resource-constrained devices due to their very high computational complexity and energy consumption.

For reducing computational and communication overhead at the user side, Naehrig *et al.* [20] suggested that instead of using FHE schemes at the user side, the user can encrypt data using AES, which is a lightweight symmetric key encryption scheme. On the cloud side, the AES encrypted data can be converted to FHE encrypted data using homomorphic decryption of AES for further secure processing. Instead of block cipher, Canteaut *et al.* [10] proposed stream cipher-based FHE-compatible Kreyvium cipher. Kreyvium cipher improves the number of homomorphic operations supported on FHE encrypted data (homomorphic capacity) obtained through homomorphic decryption. However, the homomorphic capacity of sequentially generated Kreyvium ciphertext gradually decreases. Moreover, the practical implementation of homomorphic decryption of AES [11], and Kreyvium stream cipher [10] is time-consuming. Furthermore, they do not support the homomorphic evaluation of integer arithmetic required for DT-based disease detection.

Decision tree machine learning algorithms are widely used in disease diagnosis and detection [21] such as tachycardia classification [22], genomics detection [23] and cancer detection [24]. For preserving privacy in outsourced decision-tree-based disease detection, private DTCs (PDTCs) are implemented in the encrypted domain [6]–[9]. Bost *et al.* [7] implemented PDTC using additive homomorphic Paillier encryption, quadratic residuosity cryptosystems, and fully homomorphic BGV encryption scheme. In their setting, the classifier model is private to the server. However, the computational overhead at the user side and the number of interactions between user and server are high. Ma *et al.* proposed a tree-based classifier using their additively homomorphic two trapdoor cryptosystem (TTC) [9]. Ma *et al.* also proposed a variant of the tree-based extreme gradient boosting (XGBoost) model using the TTC scheme [8]. They consider the classifier model as private to the server. However, these schemes do not preserve the medical user's privacy since it reveals disease status to the health cloud. Also, these schemes are not energy-efficient due to the use of the TTC encryption scheme. Sun *et al.* [6] constructed a PDTC using their improved FHE scheme. In their setting, the classifier model is public, and their algorithm hides only the user's input. However, computational and communication overhead at the user side is significantly high.

The summary of the major goals achieved by the private DTC (competing schemes available in the literature and the proposed scheme) for disease detection is given in Table 1. Since Zhuoran *et al.'s* PDTC [8], [9] makes use of an encryption scheme that supports only additive homomorphism, it can't fully preserve the privacy of the user as it reveals the disease status to the cloud. Sun *et al.'s* PDTC [6] does not preserve the privacy of the cloud since the secure comparison in Sun *et al.'s* PDTC reveals a classification model to the user. The existing privacy-preserving DTC can't provide energy efficiency at the user device since they make use of energy-consuming public key-based ciphers. Hence, this work aims to design a practical, energy-efficient, privacy-preserved, DTC based on energy-efficient FHE compatible cipher. FHE compatible cipher is designed to support integer addition and multiplication in the encrypted domain through fast homomorphic decryption. Specifically, our proposed PDTC-LRC is designed to achieve the following two major goals.

TABLE 1. Summary of the goals of the proposed PDTC-LRC and goals achieved by the existing private DTC.

Private DTC	User privacy	Health cloud privacy	Energy efficiency at user
Sun *et al.*'s PDTC [6]	Yes	No	No
Bost *et al.*'s PDTC [7]	Yes	Yes	No
Zhuoran *et al.*'s PDTC [8, 9]	No	Yes	No
Proposed PDTC-LRC	Yes	Yes	Yes

Security: Ensure data confidentiality of both the medical user and the health cloud during data processing and transmission operations in DT-based disease classification.

Efficiency: Achieve energy efficiency at the resource-constrained medical user without affecting the classification accuracy and speed of process while porting DT in the plain domain to that in the encrypted domain.

III. PRELIMINARIES

This section describes the encryption methods [10]–[12], and nonlinear filter generator [26] based on which FCRS is developed. Finally, the basics of private DTC [6], [7] a classifier used for disease detection, are also described.

A. RIVEST *et al.'s* SYMMETRIC ENCRYPTION SCHEME

Rivest *et al.'s* symmetric encryption scheme [12] is a Chinese remainder theorem based scheme, where two large primes (p, q) are secret keys and $n = p * q$ is a public parameter. The security of this scheme is based on factorization problem (same as that of RSA algorithm).

B. FHE-COMPATIBLE ENCRYPTION SCHEME

The FHE-compatible encryption schemes [10], [11] are proposed such that user encrypts message, m with a symmetric encryption scheme E using the secret key k and generates the ciphertext as $c' = E_k(m)$. The ciphertext c' is sent to the cloud for processing in encrypted domain. As part of initialization, the user stores the FHE encrypted secret key k as $FHE_{pk}(k)$ at cloud, where pk is the public key of the FHE scheme. The user will hold the secret keys of the FHE scheme. When cloud receives $c' = E_k(m)$, it exploits the homomorphic property of FHE scheme and recovers $FHE_{pk}(m)$ as follows,

$$c = FHE_{pk}(m) = HEL_{E^{-1}}(FHE_{pk}(k), E_k(m))$$

where HEL is the homomorphic evaluation of the decryption operation E^{-1} of symmetric encryption scheme with secret key k. After processing FHE encrypted data generated through the homomorphic decryption, the cloud sends the resulting ciphertext to the medical user at the lowest 'level'

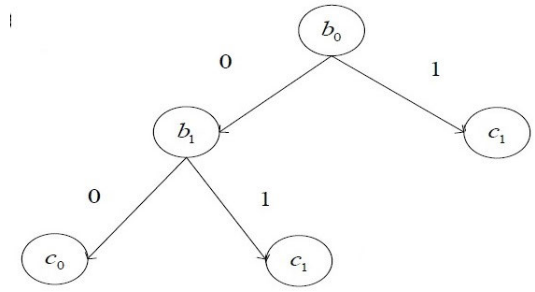


FIGURE 1. Decision tree.

of the FHE scheme to reduce computation and communication overhead for the user. The parameter 'level' of the FHE scheme determines the noise tolerance and the number of consecutive homomorphic operations supported by the scheme. The FHE-compatible scheme has the following advantages: (i) reduces storage, computation, and communication requirements of the user and (ii) privacy-preserved processing of data using additive and multiplicative homomorphism of the FHE scheme at the cloud.

C. NONLINEAR FILTER GENERATOR

Nonlinear filter generator (NFG) is a basic key stream generator for stream cipher applications consisting of a single LFSR whose output is filtered by a nonlinear function (NF) [25].

Definition 1: Let $k_n = a_0, a_1, a_2, \ldots$ is a infinite sequence generated by the LFSR. Then the sequence $k_{n+t} = a_t, a_{t+1}, a_{t+2}, \ldots$ is called the t^{th} phase shift of k_n [26].

The nonlinear filter generator (NFG) that adds second order pseudo random sequence with single order pseudo random sequence is balanced and satisfies good randomness properties if it satisfies the following condition. Let $g = k_n * k_{n+1} + k_{n+t}$ is the sequence generated by the NFG, where k_n is the infinite series generated by primitive LFSR. Then g is balanced and satisfies good randomness properties if k_{n+t} is not equal to either k_n or k_{n+1} (i.e, if $t > 1$) [26].

D. DECISION TREE CLASSIFIER (DTC)

A binary decision tree for a two-class problem (C_0 and C_1) is shown in Fig. 1, where b_i represents the result of comparison of weight and input feature at i^{th} node. The steps involved in the implementation of DTC are integer comparison at each node of the decision tree, and polynomial evaluation on the comparison results (b_is) [6], [7]. The corresponding polynomial evaluation of DTC shown in Fig.1 to obtain classification result ($Clsdct$) using b_i at each node of DTC can be formulated as in Eqn. (1).

$$Clsdct = (b_0) * C_1 + (1 - b_0) * (b_1) * C_1 + (1 - b_0) * (1 - b_1) * C_0 \quad (1)$$

For the implementation of a privacy preserved decision tree classifier, integer comparison at each node of the decision tree and polynomial evaluation of DTC should be implemented in encrypted domain [6], [7]. The polynomial evaluation of DTC in the encrypted domain requires an FHE scheme since evaluation in the plain domain involves both addition and multiplication. Hence Eqn. (2) gives the polynomial evaluation in encrypted domain corresponding to Eqn. (1).

$$\begin{aligned} FheEnc_{pk}(Clsdct) &= FheEnc_{pk}(b_0) *_c C_1 +_f \\ & FheEnc_{pk}(1 - b_0) *_f FheEnc_{pk}(b_1) *_c C_1 +_f \\ & FheEnc_{pk}(1 - b_0) *_f FheEnc_{pk}(1 - b_1) *_c C_0 \end{aligned} \quad (2)$$

where $FheEnc_{pk}$ denote the FHE encryption of data using the public key (pk) of the FHE scheme. $+_f$ and $*_f$ denote additive and multiplicative homomorphism of the FHE scheme, respectively. $*_c$ denote constant multiplication in the FHE scheme.

IV. SYSTEM SPECIFICATIONS

In this section, the system model and adversary model are defined.

A. SYSTEM MODEL

In the proposed system, a medical user wearing a body sensor network uses a smartphone to transmit encrypted personal health information (PHI) to the health cloud for secure disease diagnosis, as shown in Fig. 2. At the health cloud side, a private DT algorithm is executed to detect the disease status of the medical user. The DTC model parameters are considered as the proprietary of the health cloud due to the competitive advantage in outsourced disease detection.

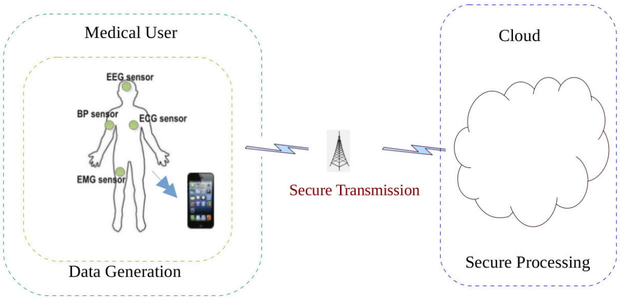


FIGURE 2. System model.

B. ADVERSAY MODEL

In this work, we assume that medical users and health cloud (internal adversaries) involved in the execution of private DT algorithms are honest and follow the steps in the algorithm correctly without manipulating the data owned by each entity. However, internal adversaries may try to learn more information than allowed by looking at the transcript of messages that they receive. Passive external adversaries can eavesdrop on the data communicated between the medical user and the cloud. The external adversary's goal is to breach the confidentiality of the PHI of the medical user. The adversaries launch different attacks such as ciphertext-only attacks

(COA), known-plaintext attacks (KPA), man-in-the-middle attacks, and spoofing attacks to deduce the keys used for encryption. The important security requirements are summarized below.

Secure the Medical Data of the User: Attackers cannot obtain the content of medical data if they eavesdrop on the communication channel and while processing data at the cloud.

Secure the Classifier Model of the Health Cloud: The intermediate data sent to the medical user by the health cloud should not reveal any information about the classifier model in the scenario where the DT classifier model parameters are proprietary of the health cloud.

V. PROPOSED FHE-COMPATIBLE RIVEST SCHEME (FCRS)

In this work, a lightweight additive homomorphic encryption scheme is proposed by modifying Rivest *et al.'s* symmetric encryption scheme (RSE) [12] for the effective implementation of outsourced disease detection. RSE is vulnerable to COA since it is not semantically secure (it always maps a plaintext to the same ciphertext). Hence, a random number is added before taking the modulus in RSE to ensure one to many mapping in the encryption scheme, thereby offering semantic security and resistance to COA. Moreover, the proposed encryption scheme can be decrypted homomorphically to support both homomorphic integer multiplication and addition required for the private DT classification. Various steps in the proposed $FCRS = (KeyGen, Encrypt, Decrypt, Add, ConstAdd)$ are outlined as below:

- FCRS.KeyGen(λ): Takes as input, the parameter λ which determines the security of the FCRS scheme. Generate three primes p, q and v and an integer L based on security parameter (λ). The p and q should be equal in length and $v \ll n = p * q$. Pick a random seed $g_{seed} = (k_{L-1}\, k_{L-2} \ldots k_0)$ such that k_i s are chosen from Z_v. The parameters p, q, v and g_{seed} are kept as secret.
- FCRS.IterkeyGen(g_{seed}, v, L): Takes as input the length L and the initial state g_{seed} of the NFG together with a secret prime v. An iteration key g_i is generated from g_{seed} based on the NFG [26]. Let $k_n = (k_i)_{i=0}^{\infty}$ be the sequence over Z_v generated by the primitive LFSR. Let the primitive feedback polynomial be $f_0 + f_1x + f_2x^2 + \ldots + f_Lx^L$. The recurrence relation of LFSR at the i^{th} instant can be expressed as given in Eqn. (3)

$$k_{L+i} = \sum_{n=1}^{L} f_n * k_{L+i-n}\, mod\, v \tag{3}$$

The following nonlinear function which takes inputs from LFSR, generates i^{th} iteration key g_i as given in Eqn. (4)

$$g_i = (k_{1+i} * k_{2+i}) + k_{5+i} \tag{4}$$

Since each $k_i \in Z_v$, the maximum value for g_i is $v^2 - v$

- FCRS.Encrypt(m_i, g_i, p, q): Takes as inputs the i^{th} message to be encrypted $m_i \in Z_e$, i^{th} iteration key g_i generated by the FCRS.IterkeyGen function and secret primes p and q generated by the FCRS.KeyGen function. Since the maximum value for g_i is $v^2 - v$, e could be any prime lower than $n - (v^2 - v)$. The ciphertext $c_i = (c_{i1}, c_{i2}) = (FCRS_p(m_i), FCRS_q(m_i))$ is generated as two shares of the message m_i based on Chinese remainder theorem (CRT) as follows.

$$(c_{i1}, c_{i2}) = ([(m_i + g_i)]_p, [(m_i + g_i)]_q).$$

- FCRS.Add($FCRS_{p,q}(m_1), FCRS_{p,q}(m_2)$): The two ciphertexts ($FCRS_p(m_1), FCRS_q(m_1)$) and ($FCRS_p(m_2)$, $FCRS_q(m_2)$) are given as input to the function FCRS.Add. The function FCRS.Add will produce an additive ciphertext ($FCRS_p(R), FCRS_q(R)$) as given below,

$$FCRS_p(R) = FCRS_p(m_1) + FCRS_p(m_2);$$
$$FCRS_q(R) = FCRS_q(m_1) + FCRS_q(m_2);$$

- FCRS.ConstAdd(c_i, t): A ciphertext $c_i = (FCRS_p(m_i)$, $FCRS_q(m_i))$ and a constant t are given as input to the function FCRS.ConstAdd. The function FCRS.ConstAdd compute $((t + (FCRS_p(m_i)), (t + (FCRS_q(m_i)))$, that is equivalent to $(FCRS_p(t + m_i)$, $FCRS_q(t + m_i))$.
- FCRS.Decrypt(c_i, g_i, p, q): Takes as input the ciphertext c_i corresponding to i^{th} message m_i, i^{th} iteration key g_i and secret primes of FCRS (p and q). From $c_i = (c_{i1}, c_{i2}) = (FCRS_p(m_i), FCRS_q(m_i))$, decrypted result ($m_i$) is obtained using CRT as follows,

$$m_i = CRT([c_{i1}]_p, [c_{i2}]_q) - (g_i).$$

where, $CRT([c_{i1}]_p, [c_{i2}]_q) = [(([c_{i1}]_p * [q^{-1}]_p * q) + ([c_{i2}]_q * [p^{-1}]_q * p))]_n$.

The proof for the correctness of the decryption operation and the homomorphism of operations in FCRS is given in the supplementary material to this article.

A. DESIGN CONSIDERATIONS FOR FCRS

The parameters for the *FCRS* need to be chosen properly to ensure the required level of security with minimum possible complexity of operations. The iteration key (g_i) needs to be developed through low complex operations to ensure efficient resource utilization at the user side. In addition, g_i needs to be derived through operations that preserve homomorphism at the cloud side. Hence, in the proposed work, it is suggested to generate g_i with a nonlinear filter generator (NFG) which will help to keep the complexity of operations low, as discussed in Section III-C.

The primitive feedback polynomial is made public to improve the homomorphic capacity and reduce the time required for homomorphic decryption. Unlike stream cipher schemes, this will not make the system vulnerable to KPA due to the generation of shares with RSE. The attacker cannot

$\mathbf{U}ser$	$\mathbf{C}loud$
$Stored\ data:$ $p, q, v, g_{seed}.$	$Stored\ data: FKV, pk, FheEnc_{pk}(v),$ $FheEnc_{pk}(n), FheEnc_{pk}([p^{-1}]_q * p),$ $FheEnc_{pk}([q^{-1}]_p * q).$

Step-I: Data preparation

(c_{i1}, c_{i2}) = FCRS.Encrypt (m_i, g_i, p, q)

$d_i = \lfloor((c_{i1} * [q^{-1}]_p * q) + (c_{i2} * [p^{-1}]_q * p))/n\rfloor$

$g_i = FCRS.IterkeyGen(g_{seed}, v, L)$

$z_i = (\sum_{n=1}^{L} f_n * k_{L+i-n})/n$

$$\xrightarrow{c_{i1}, c_{i2}, d_i, z_i}$$

Step-II: Homomorphic Decryption

Compute $FheEnc_{pk}(g_i)$ using Eqn. (9)

$FheEnc_{pk}(m_i) = (c_{i1}$
$*_c FheEnc_{pk}([q^{-1}]_p * q)) +_f (c_{i2}$
$*_c FheEnc_{pk}([p^{-1}]_q * p))$
$-_f (d_i *_c FheEnc_{pk}(n)) -_f (FheEnc_{pk}(g_i)$

Protocol 1. Homomorphic Decryption Protocol (HDP).

retrieve g_i from shares of $(m_i + g_i)$ without knowing secret parameters p, q and v.

B. HOMOMORPHIC DECRYPTION

FCRS supports decryption operations in the encrypted domain using the homomorphic properties of the FHE scheme. Hence, to make homomorphic decryption possible at cloud, FCRS.Decrypt(c_i, g_i, p, q) should be modified such that it contain only addition and multiplication operations. The decryption of *FCRS* through CRT includes modular reduction by n. However, modular reduction of a value x by n ($[x]_n$) can be realized using addition and multiplication as follows

$$[x]_n = x - \lfloor x/n \rfloor * n \tag{5}$$

Hence the decryption operation of FCRS is modified as in Eqn. (6) such that it can be realized using additive and multiplicative homomorphism.

$$CRT(c_{i1}, c_{i2}) = [(c_{i1} * [q^{-1}]_p * q) + (c_{i2} * [p^{-1}]_q * p)] - d_i * n, \tag{6}$$

where $d_i = \lfloor((c_{i1} * [q^{-1}]_p * q) + (c_{i2} * [p^{-1}]_q * p))/n\rfloor$.

Then the message m_i corresponding to (c_{i1}, c_{i2}) can also be retrieved as follows,

$$m_i = [(c_{i1} * [q^{-1}]_p * q) + (c_{i2} * [p^{-1}]_q * p)] - d_i * n - g_i \tag{7}$$

For FHE encryption through homomorphic decryption, Eqn. (7) can be realized using FHE, since it involves only addition and multiplication. Thus, the $FheEnc_{pk}(m_i)$ can be computed as given in Eqn. (8). $FheEnc_{pk}$ denotes the FHE encryption of data using public key (pk) of the medical user. Here BGV scheme [14] is used as FHE scheme. $+_f$, $-_f$ and $*_f$ denotes homomorphic addition, homorphic subtraction and homomorphic multiplication of the FHE scheme, respectively. $*_c$ denotes constant multiplication in the FHE scheme.

$$\begin{aligned} &FheEnc_{pk}(m_i) \\ &\quad = [(c_{i1}) *_c FheEnc_{pk}([q^{-1}]_p * q)] \\ &\qquad +_f [(c_{i2}) *_c FheEnc_{pk}([p^{-1}]_q * p)] \\ &\qquad -_f [(d_1) *_c FheEnc_{pk}(n)] -_f [FheEnc_{pk}(g_i)] \end{aligned} \tag{8}$$

To realize Eqn. (8), it is required to generate g_i in the encrypted domain as $(FheEnc_{pk}(g_i))$. To generate $FheEnc_{pk}(g_i)$ at cloud, the medical user initially stores FHE encrypted seed of key vector (FKV) as $\{FheEnc_{pk}(K_{L-1}), \ldots FheEnc_{pk}(K_1), FheEnc_{pk}(K_0)\}$ and $FheEnc_{pk}(v)$ at the cloud. The values of the state registers from position 0 to $(L-2)$ of LFSR in encrypted domain can be updated by the mere shifting of the FHE encrypted key vector. To update $(L-1)^{th}$ element of FKV, cloud can select the feedback connections based on publicly known primitive feedback polynomial. Then Eqn. (3) and (4) can be realized using the additive and multiplicative homomorphism of FHE scheme. For realizing the modulo v operation in Eqn. (3) based on Eqn.(5), the user has to send $z_i = \left\lfloor (\sum_{n=1}^{L} f_n * k_{L+i-n})/v \right\rfloor$ to the cloud. Thus Eqn. (4) can be computed in encrypted domain to generate $FheEnc_{pk}(g_i)$ as given Eqn. (9).

$$\begin{aligned} FheEnc_{pk}(g_i) = (&FheEnc_{pk}(k_{1+i}) *_f FheEnc_{pk}(k_{2+i})) \\ &+_f FheEnc_{pk}(k_{5+i}) \end{aligned} \tag{9}$$

A detailed description of the generation of random numbers ($g_i s$) for 128-bit security in plain domain and its corresponding regeneration in the encrypted domain ($FheEnc_{pk}(g_i)s$) can be found in the supplementary material to this article.

The disease diagnosis procedure, which requires both addition and multiplication, is done at the cloud in MHN. The data encrypted using additively homomorphic *FCRS* need to be converted to FHE encrypted data through homomorphic decryption procedure for further processing at the cloud server. Protocol 1 gives the steps involved in homomorphic decryption for converting *FCRS* encrypted data to FHE encrypted data. The user initially stores the secrets required for homomorphic decryption in the encrypted form as FKV $= \{FheEnc_{pk}(K_{L-1}),$ $\ldots FheEnc_{pk}(K_1), FheEnc_{pk}(K_0)\}$, $FheEnc_{pk}(v)$, $FheEnc_{pk}$ $([q^{-1}]_p * q)$, $FheEnc_{pk}([p^{-1}]_q * p)$ and $FheEnc_{pk}(n)$ at cloud server. For homomorphic decryption, the user sends ciphertext components c_{i1}, c_{i2} and the integers d_i and z_i to the cloud. d_i and z_i will not leak any secret information, since n and v are kept as secrets. After performing the steps in Protocol 1, the FHE encrypted data will be available in the cloud without causing privacy leakage.

The following section discusses the proposed private decision tree-based classification with low resource consumption based on the proposed FCRS.

VI. PROPOSED PRIVATE DECISION TREE-BASED CLASSIFICATION WITH LOW RESOURCE CONSUMPTION (PDTC-LRC)

This work attempts to develop an efficient and privacy-preserving decision tree-based disease detection with very low resource utilization at the medical user side (PDTC-LRC). The scheme is developed to ensure energy efficiency at medical user side while protecting the privacy of the health data of the medical user as well as the classifier model of the health cloud. The proposed FCRS cryptosystem is the basis for the development of PDTC-LRC. In PDTC-LRC, the DT model is considered to be the proprietary of the health cloud. As seen in Section III-D, the DT-based classification involves integer comparison at the nodes of the tree and the polynomial evaluation on the comparison results. Hence PDTC-LRC is developed based on energy-efficient secure integer comparison protocol and secure polynomial evaluation.

The block-level representation of PDTC-LRC is given in Fig. 3. Initially, in PDTC-LRC, the medical user sends features encrypted with proposed energy and bandwidth-efficient FCRS to the cloud. Then, PDTC-LRC executes secure integer comparison protocols (SICPs) between the medical user and the cloud to perform integer comparisons at nodes of the tree in the encrypted domain. The private inputs to the SICP are the nodal weight of the tree owned by the health cloud and the FCRS encrypted feature of the medical user corresponding to the node. In SICP, the health cloud performs homomorphic operations on nodal weights of the DT and homomorphically decrypted FCRS encrypted input features. Health cloud sends the result of homomorphic operations in the lowest level FHE encrypted form to the medical user as an intermediate result of the secure comparison. The medical user decrypts the lowest level FHE encrypted intermediate result and computes the comparison result in the plain domain. After performing SICP between the medical user and the cloud, the comparison results will be available to the medical user without compromising privacy between the two entities. The medical user sends FCRS encrypted comparison results to the cloud. Then cloud performs secure polynomial evaluation (SPE) on homomorphically decrypted comparison results to get FHE encrypted disease status. Cloud sends FHE encrypted disease status at the lowest level to the medical user. The lowest level (reduced dimension) FHE encrypted data has reduced ciphertext size (communication cost) and requires fewer computations for decryption. However, the lowest level ciphertext does not possess any homomorphic property. The resource-constrained medical user gets the disease status by decrypting low complex lowest level FHE encrypted disease status without much energy consumption.

The following section describes the proposed SICP executed between cloud and user as part of PDTC-LRC.

A. PROPOSED SECURE INTEGER COMPARISON PROTOCOLS (SICP)

Here, the health cloud service provider does not want to reveal the weight associated with each node of the DT classifier model. Also, features extracted at medical user for getting disease status are private to the medical user. Hence, integer comparison between nodal weight of the tree (wgt_i) and corresponding input feature (m_i) sent by the user needs to be done without disclosing each other's private inputs. The basic idea of comparing two integers a and b is to add the difference $(a - b)$ to an integer x, which is the power of 2 such that the most significant bit of the sum will be one if the difference is positive and will be 0 otherwise [27]. However, by knowing the difference $(a - b)$ and one of the values a will reveal the value b. Hence, the difference $(a - b)$ needs to be blinded through multiplication with a random number. Therefore the secure integer comparison protocol between medical user and cloud to compare m_i and wgt_i, is formulated as given in Protocol 2. The user sends the features encrypted with the proposed *FCRS* as $(c_{i1}, c_{i2}) = (FCRS_p(m_i), FCRS_q(m_i))$ and corresponding integers required for homomorphic decryption as (d_i, z_i) to the cloud. The cloud homomorphically decrypts $(FCRS_p(m_i), FCRS_q(m_i))$ to $FheEnc_{pk}(m_i)$ using Protocol 1. Cloud compute the difference $(w_i) -_c FheEnc_{pk}(m_i)$ and blinds the difference by multiplying it with a random number r'_i. Cloud sends $FheEnc_{pk}(e_i) = ((w_i) -_c FheEnc_{pk}(m_i))$ $*_c$ $(r'_i) +_c$ (2^x) at the lowest level to the medical user. $+_c$ $(-_c)$ denote constant addition (constant subtraction) in FHE scheme. x is chosen such that 2^x is greater than possible maximum value of $(w_i\text{-}m_i) \times r'_i$. User decrypts $FheEnc_{pk}(e_i)$ using the decryption operation of FHE scheme represented as $FHE.Decrypt(FheEnc_{pk}(e_i), s_k)$, where s_k is the secret key of the FHE scheme owned by medical user. After decryption, user obtains $e_i = (w_i\text{-}m_i) \times r'_i + 2^x$. Thus multiplication with r'_i will help to hide weights w_i from the user. If $(x+1)^{th}$ bit of e_i is 1, it indicates that $w_i \geq m_i$.

The following process in PDTC-LRC is secure polynomial evaluation (SPE) at the cloud. SPE is performed on the FCRS encrypted comparison results sent by the user.

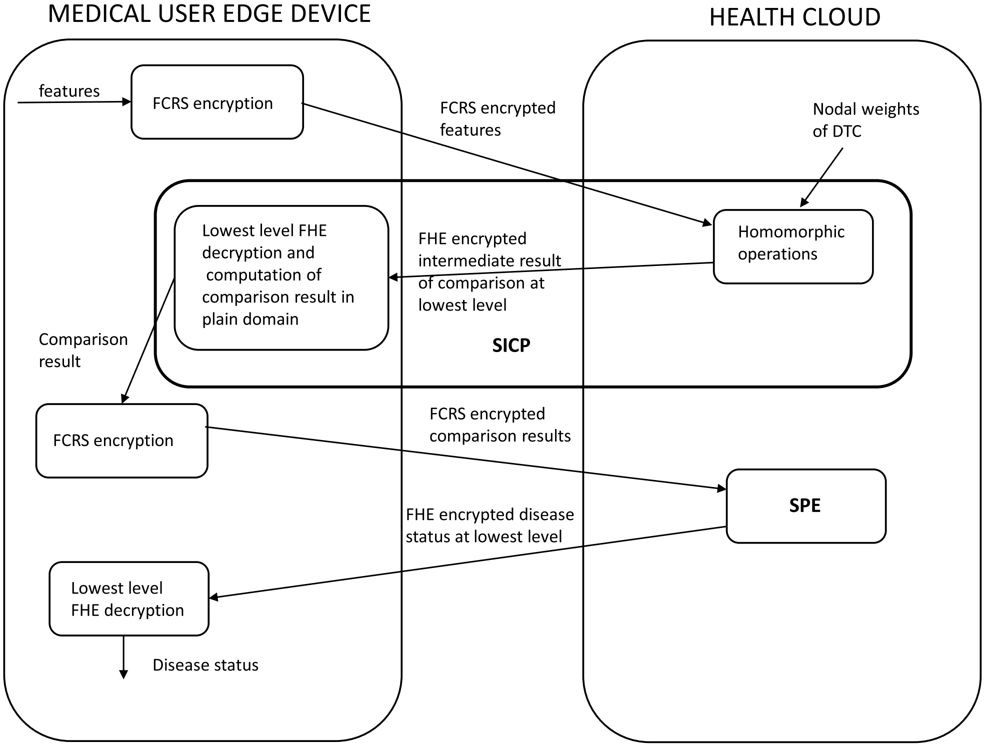

MEDICAL USER EDGE DEVICE
HEALTH CLOUD
features
FCRS encryption
FCRS encrypted features
Nodal weights of DTC
Homomorphic operations
Lowest level FHE decryption and computation of comparison result in plain domain
FHE encrypted intermediate result of comparison at lowest level
SICP
Comparison result
FCRS encryption
FCRS encrypted comparison results
SPE
FHE encrypted disease status at lowest level
Lowest level FHE decryption
Disease status

FIGURE 3. Block-level representation of the proposed PDTC-LRC.

User

Stored data :
p, q, v, g_{seed}.

Cloud

$Stored\ data: FKV, pk, FheEnc_{pk}(v),$
$pk, FheEnc_{pk}(n), FheEnc_{pk}([p^{-1}]_q * p),$
$FheEnc_{pk}([q^{-1}]_p * q).$

Step-I: Data preparation
Perform Data preparation step
of HDP with input m_i

$\xrightarrow{c_{i1}, c_{i2}, d_i, z_i}$

Step-II: Homomorphic operations
Perform Homomorphic decryption
step of HDP to get $FheEnc_{pk}(m_i)$

Choose a random number r_i'
$FheEnc_{pk}(e_i) = (w_i$
$-_c FheEnc_{pk}(m_i)) *_c (r_i')$
$+_c (2^x)$

$\xleftarrow{FheEnc_{pk}(e_i) at\ lowest\ level}$

Step-III: Comparison result
$e_i = FHE.Decrypt(FheEnc_{pk}(e_i), s_k)$
Comparison Result, $b_i = x_{i_{(x+1)}}$=$(x+1)^{th}$ bit
of e_i., where $e_{i_{(x+1)}} = 1 \implies w_i \geq m_i$

Protocol 2. Secure integer comparison protocol (SICP).

B. SECURE POLYNOMIAL EVALUATION (SPE)

In PDTC-LRC, after performing SICP, the medical user sends the *FCRS* encrypted comparison results $(c_{bi1}, c_{bi2}) = (FCRS_p(b_i), FCRS_q(b_i))$ and corresponding integers required for homomorphic decryption as (d_{bi}, z_{bi}) to the cloud for the evaluation of polynomial in the encrypted domain. As described in Section III-D, the evaluation of the polynomial form of the DTC requires both multiplication and addition performed on comparison results (b_i) in the plain domain for finding the disease class. Hence, for secure polynomial evaluation, cloud homomorphically converts *FCRS* encrypted comparison results sent by the medical user to FHE encrypted comparison results ($FheEnc_{pk}(b_i)$) and evaluate polynomial in the encrypted domain (e.g., Eqn. (2)). The next section describes the proposed PDTC-LRC protocol developed based on SICP and SPE.

C. DETAILED DESCRIPTION OF THE PROPOSED PDTC-LRC PROTOCOL

In this section, the proposed PDTC-LRC developed using SICPs and SPE is described. The proposed protocol for PDTC-LRC is given in Protocol 3. The medical user stores FHE encrypted secret parameters of the FCRS required for homomorphic decryption at the health cloud. The medical user sends M number of encrypted vital parameters ($\{c_{i1}, c_{i2}, d_i, z_i\}_{0:M}$) to health cloud. The medical user and health cloud execute secure integer comparison protocol described in Protocol 2 for Nd number of nodes in the DTC. The medical user sends the FCRS encrypted comparison results to the health cloud. At health cloud, SPE is performed to get classification results in the encrypted domain as $FheEnc_{pk}(C)$. Health cloud sends $FheEnc_{pk}(C)$ to medical user. The medical user decrypts the ciphertext to get his disease status. The health cloud can send $FheEnc_{pk}(C)$ at the lowest level (reduced dimension) so that the ciphertext size (communication cost) and computations required for decryption can be minimized.

VII. SECURITY AND PRIVACY ANALYSIS

First, this section analyzes the security of the medical data and FCRS keys against various attacks during transmission, storage, and processing operations. Further, the security analysis to verify the privacy of the classifier model parameters is presented. Finally, a formal privacy analysis of PDTC-LRC protocol is performed.

A. SECURITY OF MEDICAL DATA WHILE TRANSMITTING, STORING AND PROCESSING

The security of the medical data is preserved using FCRS and FHE (BGV) encryption schemes while transmitting, storing and processing medical data. FCRS encrypted medical data is converted to FHE encrypted medical data through homomorphic decryption operation. The security of FHE scheme (BGV scheme) relies on the hardness of ring learning with error (RLWE) problems [32]. The security of the proposed encryption scheme (FCRS) is quantified by the difficulty in solving for m_i given $c_{i1} = [m_i + g_i]_p$ and $c_{i2} = [m_i + g_i]_q$ and the integers (d_i, z_i) for modulo operation when the prime numbers (p, q, v) and initial seed of LFSR $((k_{L-1}, k_{L-2}, \ldots, k_0))$ are unknown. The number of primes less than $2^l - 1$ is approximately, $j_l = \dfrac{2^l - 1}{ln(2^l - 1)}$ [33]. The security of FCRS is analyzed in terms of ciphertext-only attack, known plaintext attack and semantic security.

1) CIPHERTEXT-ONLY ATTACK

Since the ciphertext do not leak properties of the plaintext, ciphertext only attack against FCRS is the Brute force attack itself. The resistance of the proposed encryption scheme against brute force attack is dependent on the effective keyspace considering the number of possible values of $p, q, (k_{L-1}, k_{L-2}, \ldots, k_0)$ and v. So the total number of trials required to find the secrets is given by, $T = j_{l_p} * j_{l_p} * j_{l_v} * (2^L)^{l_v}$ where j_{l_p} is the number of possibilities for selecting the prime numbers p and q of length l_p and j_{l_v} is the number of possibilities for selecting the prime number v of length l_v bits. The number of possible values for Key vector are $((2^L)^{l_v})$. Therefore when p and q are 40-bit primes, v is 15 bit prime and $L = 8$, then the attack complexity is 2^{204}.

2) KNOWN-PLAINTEXT ATTACK

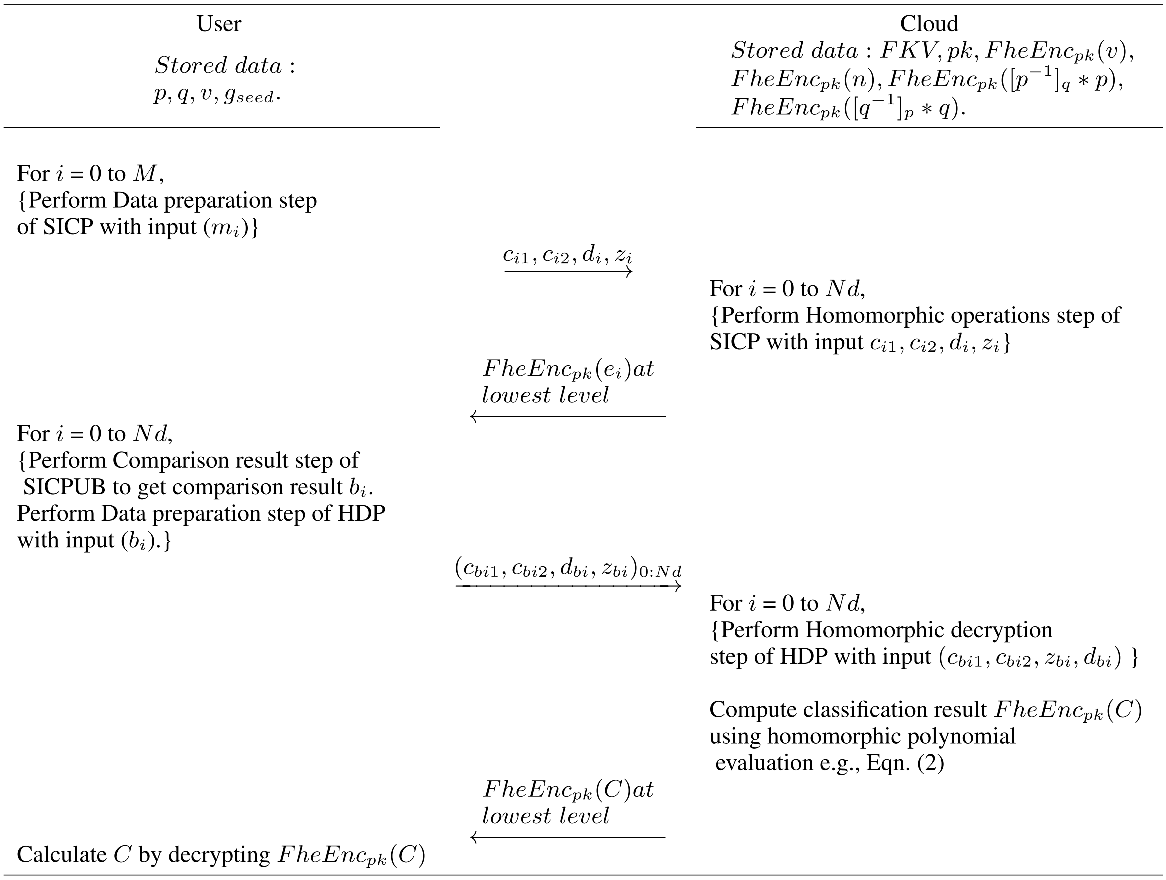
When the attacker (eavesdropper who can eavesdrop on the gateway transmissions or cloud) possesses some ciphertext-plaintext pair, he can mount a KPA to retrieve the secret parameters. Here, KPA against stream cipher cannot be mounted successfully because by knowing ciphertext-plaintext pair, the attacker will not be able to retrieve g_i since the share generating parameters of *FCRS* (p, q) are kept secret. In KPA against the proposed scheme, the attacker tries to retrieve the secret parameters $p, q, (k_{L-1}, k_{L-2}, \ldots, k_0)$ and v with known m_i, c_{i1}, c_{i2}, d_i and z_i. By knowing z_i attacker cannot extract any information about g_i, since z_i is the integer division of y_i by v (I.e., $\lfloor y_i/v \rfloor$) where y_i is sum of the elements of Key vector specified by feedback polynomial $(\left\lfloor (\sum_{n=1}^{L} f_n * k_{L+i-n})/v \right\rfloor)$ as detailed in section V-B. For all values ranging from $(z_i * v + 0$ to $z_i * v + v - 1)$ the z_i will have the same value, ensuring secrecy. So, by knowing z_i, attacker cannot find $[y_i]_v$ which ranges from 0 to $v - 1$. Since cloud possess only FHE encryption of y_i, v and Key vector (i.e, $FheEnc_{pk}(y_i)$, $FheEnc_{pk}(v)$ and FKV), cloud cannot extract any information about g_i by knowing z_i. Hence, the only way to retrieve secrets from known values of m_i, c_{i1}, c_{i2}, d_i and z_i is to solve the Eqn. (10) for all possibilities of random numbers generated using NFG.

$$\begin{aligned} m_1 &= [c_{11} * [q^{-1}]_p * q + c_{12} * [p^{-1}]_q * p] - (d_1 * n) - g_1 \\ m_2 &= [c_{21} * [q^{-1}]_p * q + c_{22} * [p^{-1}]_q * p] - (d_2 * n) - g_2 \end{aligned} \tag{10}$$

where, $n = p * q$.

The steps involved in KPA can be described as given below:

1) Pick 2 secrets m_1 and m_2 and their corresponding shares (c_{11}, c_{12}) and (c_{21}, c_{22}) respectively.
2) Guess one set of secret key values as prime number v and the initial states of the v-ary LFSR.
3) Solve the set of Eqn.s in Eqn. (10) for variables p and q.
4) Repeat step (3) for all possible values of prime number v and the initial states of the v-ary LFSR and tabulate the possible solutions of p and q.
5) Pick a new secret and its corresponding shares.



User

$Stored\ data:$
$p, q, v, g_{seed}.$

Cloud

$Stored\ data: FKV, pk, FheEnc_{pk}(v),$
$FheEnc_{pk}(n), FheEnc_{pk}([p^{-1}]_q * p),$
$FheEnc_{pk}([q^{-1}]_p * q).$

For i = 0 to M,
{Perform Data preparation step of SICP with input (m_i)}

$\xrightarrow{c_{i1}, c_{i2}, d_i, z_i}$

For i = 0 to Nd,
{Perform Homomorphic operations step of SICP with input c_{i1}, c_{i2}, d_i, z_i}

$\xleftarrow{FheEnc_{pk}(e_i)at\ lowest\ level}$

For i = 0 to Nd,
{Perform Comparison result step of SICPUB to get comparison result b_i.
Perform Data preparation step of HDP with input (b_i).}

$\xrightarrow{(c_{bi1}, c_{bi2}, d_{bi}, z_{bi})_{0:Nd}}$

For i = 0 to Nd,
{Perform Homomorphic decryption step of HDP with input $(c_{bi1}, c_{bi2}, z_{bi}, d_{bi})$ }

Compute classification result $FheEnc_{pk}(C)$ using homomorphic polynomial evaluation e.g., Eqn. (2)

$\xleftarrow{FheEnc_{pk}(C)at\ lowest\ level}$

Calculate C by decrypting $FheEnc_{pk}(C)$

Protocol 3. Private Decision Tree Classification (PDTC-LRC).

6) Perform trial and error method with all possible solutions of p and q obtained through step 4 to retrieve the actual p and q.

The number of possibilities of secret parameters for generating random numbers in LFSR are $j_{l_v} * (2^L)^{l_v}$ where j_{l_v} is the number of possibilities of selecting the prime number v of length l_v bits. The possibilities of initial seed of v-ary LFSR are $(2^L)^{l_v}$. In step (4) and (5), the attacker has to try $j_{l_v} * (2^L)^{l_v}$ possible values. Therefore, the total keyspace for KPA is $2 * j_{l_v} * (2^L)^{l_v}$. Therefore, if v is 15 bit prime and $L = 8$, then the complexity of the KPA is 2^{132}.

3) SEMANTIC SECURITY

An unpredictable pseudo-random generator (PRG) is secure, and a cipher that adds secure PRG is semantically secure [34]. Here, the random numbers are generated using the NFG given in Section III-C which is balanced and unpredictable. Also, KPA will not reveal any portion of random sequences (Section VII-A2). As the random number generated using a secure PRG is added to the message in this scheme, FCRS achieves semantic security. I.e., given two plaintexts of equal length and their respective FCRS ciphertexts, an attacker cannot correctly determine the ciphertext-plaintext pair.

B. SECURITY OF FCRS SECRET KEYS WHILE TRANSFERRING TO CLOUD/STORING AT CLOUD

The user initially sends the secret key for FCRS to the cloud securely over the channel after encrypting the secret key using FHE of the user as FKV = $\{FheEnc_{pk}(K_{L-1}), \ldots FheEnc_{pk}(K_1), FheEnc_{pk}(K_0)\}$, $FheEnc_{pk}(v)$, $FheEnc_{pk}([q^{-1}]_p * q)$, $FheEnc_{pk}([p^{-1}]_q * p)$ and $FheEnc_{pk}(n)$. The user sends FHE encrypted secret keys of FCRS from his personal device only once during the secret key updation time. Cloud server stores these encrypted secret keys of FCRS and perform homomorphic decryption of the FCRS encrypted messages using these FHE encrypted keys when the user requests for disease detection. Here $FheEnc_{pk}()$ is the BGV encryption function, where pk is the public key of the user for the BGV scheme. The secret key sk corresponding to pk is known only to the user. While transferring FCRS secret keys to the cloud over the channel, an eavesdropper can try to mount man-in-the-middle attack or spoofing attacks. In the proposed scheme, a man in the middle or spoofer gets only the FHE encrypted values corresponding to FCRS secret keys. As the FHE scheme (BGV scheme) offers security based on the hardness of ring learning with error (RLWE) problems, a man in the middle or spoofer cannot extract secret keys of the FCRS scheme by breaking the FHE. The cloud also cannot extract secret keys of FCRS due to the security offered by the BGV scheme.

C. PRIVACY OF CLASSIFIER MODEL PARAMETERS

While performing disease classification using proposed PDTC-LRC, the health cloud sends the intermediate result $FheEnc_{pk}(e_i) = ((w_i) -_c FheEnc_{pk}(m_i)) *_c (r_i') +_c (2^l)$ at lowest level of FHE encryption to medical user to get comparison result as given in Section VI-A. Even if the medical user decrypt the intermediate result, he will not get the model parameter w_i of the classifier, since cloud blinds the difference (w_i-m_i) by multiplying it with a random number r_i'. Thus multiplication with r_i' will help to hide weights w_i from the user.

D. FORMAL ANALYSIS ON PRIVACY OF MEDICAL USER AND CLOUD WHILE EXECUTING PDTC-LRC PROTOCOL

The simulation model (real vs. ideal) defined in secure two-party protocols for semi-honest adversaries [8], [35] is used to formalize privacy analysis of PDTC-LRC protocol. A protocol is said to be privacy-preserving if each party's view in the protocol (Π) execution can be simulated only when its input and output are given. Let $REAL_{\Pi,Adv_Y}$ be the real view of Party Y with input y when interacting with Party X with input x. Party X's privacy can be guaranteed if there exists a simulator Sim_Y such that for any x, $Sim_Y(y, f(x, y))$ can generate a view ($IDEAL_{f,Sim_Y}$) indistinguishable from the Y's view in the execution of the real protocol that is

$$IDEAL_{f,Sim_Y}(x, y) \overset{c}{\equiv} REAL_{\Pi,Adv_Y}(x, y)$$

where f is the function that is computed using Π.

The proposed PDTC-LRC is built based on HDP and SICP. Initially, the privacy preservation of HDP and SICP are analyzed. Then privacy analysis of PDTC-LRC is performed based on the privacy preservation of HDP and SICP.

Theorem 1: The proposed homomorphic decryption protocol (HDP) is secure against semi-honest cloud based on the security of FCRS and BGV, which can resist the distinguishment of user's medical data.

The proof of Theorem 1 is given in the supplementary material to this article.

Theorem 2: The proposed secure integer comparison protocol SICP is secure against semi-honest cloud and medical users based on the security of FCRS and BGV, which can resist the distinguishment of user's medical data and DTC parameters of the health cloud.

The proof of Theorem 2 can be found in the supplementary material to this article.

Theorem 3: The proposed PDTC-LRC is secure against semi-honest cloud and medical users based on the security of FCRS and BGV, which can resist the distinguishment of user's medical data and DTC parameters of the health cloud.

The proof of Theorem 3 is given in the supplementary material to this article.

VIII. IMPLEMENTATION RESULTS AND EFFICIENCY ANALYSIS

In this section, the experimental settings and two real-life applications are first detailed. Then experimental results are furnished to illustrate the improved performance of FCRS, SICP, and PDTC-LRC compared to the state-of-the-art schemes.

A. EXPERIMENTAL SETTINGS

The experimental environment is set up as follows: at the cloud side, the operating system is Ubuntu 16.04, featuring Intel Xeon(R) CPU E5-1620 v2 processor, running at 3.6 GHz, with 7.7 GB memory, and at the user side, Ubuntu MATE operating system runs on ARMv8 processor in Raspberry Pi 3B+ board running at 1.4 GHz, with 1 GB memory. A USB voltage-current meter (xcluma BE-001346) is used to measure the energy required for executing programs using the Raspberry Pi 3B+ board. The experiments are conducted for the security parameter, $\lambda = 128$.

B. REAL LIFE APPLICATIONS

For experiments, real-life applications in the detection of two diseases are considered: (i) Atrial fibrillation (AF) and (ii) Angiographic disease (AG). AF is a common arrhythmia among elderly people, and it is characterized by irregularly irregular RR intervals, the absence of P-waves, and fibrillatory waves. AF detection is a two-class problem (i.e., class = AF, No AF). For Atrial Fibrillation (AF) disease detection 12-lead ECG CPSC dataset [36] is considered. The features RR irregularity measure, P-wave evidence score-I, P-wave evidence score-II, and P-wave evidence score-III are extracted from the 12-lead ECG dataset [37]. The extracted features are applied to the WEKA interface to train the DTC for AF.

AG indicates diameter narrowing of the heart's blood vessels, which requires immediate medical care. AG detection outputs either *AG* or *NoAG*, where *AG* and *NoAG* correspond to "greater than 50% diameter narrowing" and "less than 50% diameter narrowing" of the heart's blood vessels, respectively. The features available in the angiographic disease UCI dataset [38] are applied to the WEKA interface to train the DTC for AG.

C. COMPARISON OF PROPOSED FCRS AND ENCRYPTION SCHEMES USED FOR PRIVACY PRESERVED MEDICAL APPLICATIONS

The encryption schemes used at the user side for the privacy-preserving medical applications are analyzed. For the proposed FCRS, p and q are chosen as 32-bit prime numbers, v is selected as a 15-bit prime number, and L is set as 8 to achieve the 128-bit security requirement in MHN. Table 1 shows the comparison of the proposed FCRS with encryption schemes used for privacy-preserved medical applications for the security parameter, $\lambda = 128$. Here, evaluation metrics are computational time and energy utilization

TABLE 2. Comparison of computational and communication overhead between proposed *FCRS* and encryption schemes used for privacy preserved medical applications [6], [8], [9], [13], [15]–[19], [39] ($\lambda = 128$).

Encryption scheme	Encryption		Decryption		Ciphertext Size	Transmission Energy
	Time (ms)	Energy (mJ)	Time (ms)	Energy (mJ)	(bits)	(mJ)
Proposed $FCRS$	.021	0.0043	.028	0.0057	100	0.0003
MSSS [17]	0.02	0.0041	0.025	0.0051	64	0.0002
MRSE([13])	0.014	0.0028	0.129	0.0266	9216	0.031
SHE([18])	0.0171	.0035	0.0173	.0035	128	0.0004
SFH-DEM([19])	0.513698	0.1062	0.012	0.0024	768	.002
BGV scheme ([15])	188.9	175.79	97.4	90.64	1448098	4.9
Sun's FHE scheme ([6])	188.9	175.79	97.4	90.64	1448098	4.9
Paillier ([16])	592.6	551.47	18.9	17.58	6144	0.02
TTC ([8], [9])	301.3	280.66	106.9	99.48	6144	0.02
GM cryptosystem([39]) (32 bits)	4.48	.92	304	282.89	98304	0.335

at the user side for encryption, decryption, and transmission operations. Energy utilization at the user side for encryption, decryption and transmission operations in FCRS encryption is much lesser than that of FHE schemes (Sun's FHE [6], and HElib [15]). Also, computational time, ciphertext expansion, and energy utilization with proposed FCRS encryption are much lesser than the corresponding parameter values of additive homomorphic public-key encryption schemes (Paillier, GM, TTC) used in [7], [8]. Though the performance of the proposed FCRS in terms of energy efficiency is comparable with SHE, MRSE, and MSSS schemes these schemes possess only additive homomorphism. They do not support homomorphic decryption at the cloud, which is required for the privacy-preserving DTC-based medical diagnosis. Similarly, the classification performed with symmetric key-based FHE scheme, SFH-DEM cannot preserve the privacy of the health cloud.

D. COMPARISON OF PROPOSED SICP AND EXISTING SECURE INTEGER COMPARISON PROTOCOLS

The performance of the proposed secure integer comparison protocol (SICP) is compared with that of the existing secure integer comparison protocols Bost *et al.'s* secure comparison protocol (BSCP) [7], Ma *et al.'s* secure comparison protocol (ZhSCP) [8], [9], and Sun *et al.'s* secure integer comparison protocol (SunSCP) [6] available in literature for the security parameter, $\lambda = 128$. Here, evaluation metrics are computational overhead at the user (computational time and energy consumed for computation), transmission overhead at the user (Number of bits communicated (kB) and energy consumption for communication), total energy (energy consumption for computation and communication), computational time at cloud, total computational time (time at user+ time at cloud), number of communication rounds and model privacy.

Table 2 shows that the proposed SICP, ZhSCP, and SunSCP require only one round of communication between the medical user and cloud, whereas BSCP requires three rounds of communication. The proposed SICP, BSCP, and ZhSCP preserve the privacy of both the medical user and the health cloud. The proposed SICP consumes less time at the user than BSCP, SunSCP and ZhSCP. The computational time at the cloud is more in the proposed SICP when compared to SunSCP and ZhSCP. However total time required for the execution of SICP is lesser than SunSCP, BSCP and ZhSCP. The communication overhead of our scheme is lesser than that of BSCP and SunSCP. Moreover, the total energy consumption for the proposed SICP is much lesser than BSCP, SunSCP and ZhSCP. Hence, the proposed SICP outperforms existing schemes in terms of user resource utilization.

TABLE 4. Comparison of DTC in plain domain and encrypted domain.

Classifier	Accuracy	TP rate	FP rate
AG-DTC	86.8 %	0 .868	0.14
AG-PDTC-LRC	86.8 %	0.868	0.14
AF-DTC	93.15%	0 .932	0.086
AF-PDTC-LRC	93.15%	0 .932	0.086

E. ACCURACY OF PROPOSED PDTC FOR AF AND AG DETECTION

For the proposed PDTC-LRC model, training is done at the health cloud (healthcare service provider), which is part of a hospital with the medical data silos. After the training procedure, the DTC model will be available at the health cloud. The decision tree classifier for Angiographic disease diagnosis (AG-DTC) and Atrial fibrillation diagnosis (AF-DTC) are trained using WEKA at the health cloud. The evaluation metrics used at the training and testing process of the DTC are detection accuracy, true positive (TP) rate, and false positive (FP) rate. AG-DTC and AF-DTC give an accuracy of 86.8% with tree depth = 5 and 93.31% accuracy with tree depth = 2, respectively, for the plain domain input features. For the privacy preservation of the user's input features and cloud's DTC model, the classification algorithm is performed as given in the PDTC-LRC protocol with encrypted input features. For the private DTC-based AF detection (AF-PDTC-LRC), steps involved in PDTC-LRC can be executed with $M = 2$ and $Nd = 2$ to achieve an accuracy of 93.31%. For the private DTC-based AG detection (AG-PDTC-LRC), steps involved in PDTC-LRC can be executed with $M = 7$ and $Nd = 9$ to achieve an 86.8% accuracy. While porting the plain domain operations in DTC to PDTC-LRC, accuracy is not changed, as shown in Table 4.

TABLE 3. Comparison between proposed SICP and existing secure integer comparison protocols [6]–[9] ($\lambda = 128$).

Protocol	User					Cloud	Total time (ms)	Number of rounds	Classifier model
	Computational overhead		Communication overhead		Total energy (mJ)	Computational overhead			
	Time (ms)	Energy (mJ)	No. of bits (kB)	Energy (mJ)		Time (ms)			
Proposed SICP	31	28.84	15.6	.435	29.275	82	113	1	private
BSCP [7]	626.1	582.64	16.87	0.471	583.11	181.85	807.95	3	private
ZhSCP [8], [9]	408.5	380.15	1.5	0.041	380.19	60.5	469	1	private
SunSCP [6]	403.8	375.77	461.29	12.88	388.65	30.2	434	1	public

F. COMPARISON OF PROPOSED PDTC-LRC AND EXISTING PRIVACY-PRESERVING DT-BASED CLASSIFIERS

The efficiency of the proposed PDTC-LRC (AF-PDTC-LRC and AG-PDTC-LRC) is compared with that of existing schemes Bost *et al.'s* private decision tree classifier (PDTCB) [7], Ma *et al.'s* private decision tree classifier (PDTCZh) [8], [9] scheme, and Sun *et al.'s* private decision tree classifier (PDTCS) [6] for the security parameter, $\lambda = 128$. Here, evaluation metrics are computational overhead at the user (computational time and energy consumption for computation), transmission overhead at user (Number of bits communicated (kB) and energy consumption for communication), total energy at the user (energy consumption for computation and communication), computational time at cloud, total computational time (time at user + time at cloud), energy consumed at user for 1 hour (PDTC is formed every 2 minutes) and battery life of the user device.

Table 5 shows that the computational overhead at the user in existing schemes is more than that in the proposed PDTC-LRC for AG and AF. The total time required for the execution of PDTC-LRC for AF and AG is lesser than existing schemes that preserve the privacy of both the medical user and cloud. Though the total computational time for the execution of PDTC-LRC for AG is comparable with that of PDTCS for AG, PDTCS cannot achieve energy efficiency at the user and cannot preserve the privacy of the health cloud.

The limitation of the proposed PDTC-LRC is the slight increase in the computational time at the cloud due to homomorphic decryption. The comparison in terms of computational time is given in Table 5. It can be noted that even with this increase, the overall time is well within a few seconds or a fraction of seconds depending upon the model of the disease being detected.

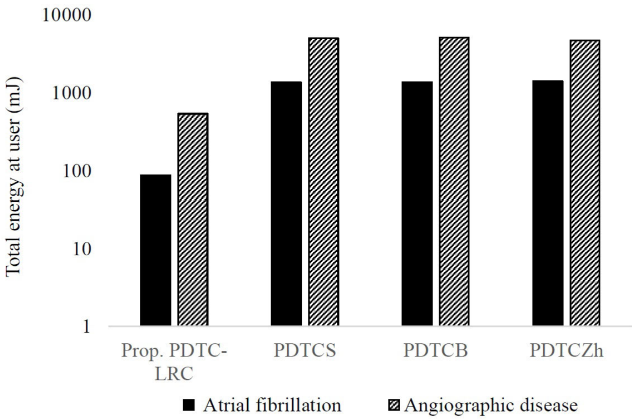


FIGURE 4. Comparison of energy usage at user for proposed PDTC-LRC and existing schemes.

The battery life for running PDTC-LRC is estimated based on a 10.78 Wh battery [40]. Results show that the battery life is significantly improved for the proposed PDTC-LRC compared to existing schemes. Fig. 4 shows that the energy consumption at the user side for the proposed PDTC-LRC for AF and AG is much lesser than those of existing schemes.

TABLE 5. Comparison among proposed PDTC-LRC, PDTCS, PDTCB and PDTCZh [6]–[9] ($\lambda = 128$).

Protocol	Disease	User				Cloud	Total time (ms)	Energy for 1 Hour (J)	Battery lifetime (Hours)
		Computational overhead		Communication overhead		Computational time			
		Time (ms)	Energy (mJ)	No. of bits (kB)	Energy (mJ)	(ms)			
Proposed AF-PDTC-LRC	AF	93.04	86.58	46.8	1.3	297	390.04	2.63	14755.89
AF-PDTCB [7]		1481.26	1378.46	241.12	6.7	481.7	1962.96	41.55	934
AF-PDTCZh [8], [9]		1522	1416.37	5.25	.146	121	1643	42.48	913.55
AF-PDTCS [6]		1414.5	1316.3	1853	51.7	74.8	1489.3	41.04	945.61
Proposed AG-PDTC-LRC	AG	576	536.02	124.8	3.48	4803	5379	16.18	2398.5
AG-PDTCB [7]		5458.8	5079.95	1438.875	40.1	1912.85	7371.65	153.6	252.65
AG-PDTCZh [8], [9]		5062	4710.69	16.5	.46	423.5	5485.5	141.33	274.59
AG-PDTCS [6]		5141	4784.2	9847.57	275	89	5230	151.77	255.7

IX. CONCLUSION

In this paper, a private decision tree-based disease detection scheme with low resource consumption at the user side (PDTC-LRC) is proposed for mobile healthcare networks. A lightweight FHE-compatible symmetric key encryption scheme FCRS and energy-efficient SICP are developed to facilitate private disease classification at the cloud. The SICP is developed based on the FCRS by considering the decision tree's weights as confidential to the health cloud. The security and privacy analysis demonstrate that the proposed schemes can resist possible attacks and preserve the data privacy of the user and health cloud. Results of experiments conducted on the Raspberry Pi 3B+ board indicate that the computational and transmission energy for the medical user is significantly reduced compared to current schemes. Reduced energy consumption helps in improving the battery life of smart devices at the user side. Moreover, privacy preservation is achieved with fast disease classification and the same accuracy as DTC in the plain domain. As COVID-19 demands real-time and remote health monitoring, hospitals can make use of the services of health clouds to provide privacy-preserving remote disease detection by adopting the proposed PDTC-LRC without any privacy concern for the user as well as the hospital/health cloud. The classifier model corresponding to the detection of two heart diseases (Angiographic disease diagnosis (AG-DTC) and Atrial fibrillation diagnosis (AF-DTC)) are given in this paper. But it can be extended by including DTC corresponding to many other common diseases. Once the user registers to the remote services through the hospital, he can get the status of his disease from anywhere anytime using a smartphone with medium features in terms of battery capacity, computing power, and RAM. Therefore the proposed schemes are highly suitable for the resource-constrained MHN users to utilize cloud services for disease detection without compromising data security. Future work will address the challenge of establishing energy-efficient and privacy-preserving neural networks based on the FHE-compatible ciphers.

REFERENCES

[1] S. Alex, D. P. Pattathil, and D. K. Jagalchandran, "SPCOR: A secure and privacy-preserving protocol for mobile-healthcare emergency to reap computing opportunities at remote and nearby," *IET Inf. Secur.*, vol. 14, no. 6, pp. 670–682, Nov. 2020.

[2] X. Wang, L. Bai, Q. Yang, L. Wang, and F. Jiang, "A dual privacy-preservation scheme for cloud-based eHealth systems," *J. Inf. Secur. Appl.*, vol. 47, pp. 132–138, Aug. 2019.

[3] M. A. Sahi, H. Abbas, K. Saleem, X. Yang, A. Derhab, M. A. Orgun, W. Iqbal, I. Rashid, and A. Yaseen, "Privacy preservation in e-Healthcare environments: State of the art and future directions," *IEEE Access*, vol. 6, pp. 464–478, 2018.

[4] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017, doi: 10.1109/JIOT.2017.2694844.

[5] W. Trappe, R. Howard, and R. S. Moore, "Low-energy security: Limits and opportunities in the Internet of Things," *IEEE Security Privacy*, vol. 13, no. 1, pp. 14–21, Jan. 2015.

[6] X. Sun, P. Zhang, J. K. Liu, J. Yu, and W. Xie, "Private machine learning classification based on fully homomorphic encryption," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 2, pp. 352–364, Apr./Jun. 2018.

[7] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2015, pp. 1–14.

[8] Z. Ma, J. Ma, Y. Miao, X. Liu, K.-K.-R. Choo, R. Yang, and X. Wang, "Lightweight privacy-preserving medical diagnosis in edge computing," *IEEE Trans. Services Comput.*, early access, Jun. 24, 2020, doi: 10.1109/TSC.2020.3004627.

[9] Z. Ma, J. Ma, Y. Miao, and X. Liu, "Privacy-preserving and high-accurate outsourced disease predictor on random forest," *Inf. Sci.*, vol. 496, pp. 225–241, Sep. 2019.

[10] A. Canteaut, S. Carpov, C. Fontaine, T. Lepoint, M. Naya-Plasencia, P. Paillier, and R. Sirdey, *Stream Ciphers: A Practical Solution for Efficient Homomorphiccipher-Text Compression* (Lecture Notes in Computer Science), vol. 9783, T. Peyrin, Ed. Berlin, Germany: Springer, 2016, pp. 313–333.

[11] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the AES circuit," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 7417, R. Safavi-Naini and R. Canetti, Eds. Berlin, Germany: Springer, 2012, pp. 850–867.

[12] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," in *Foundations of Secure Computation*. New York, NY, USA: Academic, 1978, pp. 169–179.

[13] A. C.-F. Chan, "Symmetric-key homomorphic encryption for encrypted data processing," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–5.

[14] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proc. 3rd Innov. Theor. Comput. Sci. Conf. (ITCS)*, 2012, pp. 309–325.

[15] S. Halevi and V. Shoup. (2013). *Design and Implementation of a Homomorphic-Encryption Library*. [Online]. Available: https://github.com/shaih/HElib

[16] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Cham, Switzerland: Springer, 1999, pp. 223–238.

[17] V. S. Lakshmi and P. P. Deepthi, "Collusion resistant secret sharing scheme for secure data storage and processing over cloud," *J. Inf. Secur. Appl.*, vol. 60, Aug. 2021, Art. no. 102869, doi: 10.1016/j.jisa.2021.102869.

[18] C. Guo, P. Tian, and K.-K. R. Choo, "Enabling privacy-assured fog-based data aggregation in E-healthcare systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 1948–1957, Mar. 2021.

[19] J. Chen, J. Zhou, Z. Cao, A. V. Vasilakos, X. Dong, and K.-K.-R. Choo, "Lightweight privacy-preserving training and evaluation for discretized neural networks," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2663–2678, Apr. 2020, doi: 10.1109/JIOT.2019.2942165.

[20] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical," in *Proc. CCSW, ACM*, 2011, pp. 113–124.

[21] C. Kingsford and S. L. Salzberg, "What are decision trees?" *Nature Biotechnol.*, vol. 26, no. 9, pp. 1011–1013, Sep. 2008.

[22] O. Perlman, A. Katz, G. Amit, and Y. Zigel, "Supraventricular tachycardia classification in the 12-lead ECG using atrial waves detection and a clinically based tree scheme," *IEEE J. Biomed. Health Informat.*, vol. 20, no. 6, pp. 1513–1520, Nov. 2016.

[23] M. K. Leung, A. Delong, B. Alipanahi, and B. J. Frey, "Machine learning in GenomicMedicine: A review of computational problemsand data sets," *Proc. IEEE*, vol. 104, no. 1, pp. 176–197, Jan. 2016.

[24] C.-J. Tseng, C.-J. Lu, C.-C. Chang, and G.-D. Chen, "Application of machine learning to predict the recurrence-proneness for cervical cancer," *Neural Comput. Appl.*, vol. 24, no. 6, pp. 1311–1316, May 2014.

[25] J. D. Golic, A. Clark, and E. Dawson, "Generalized inversion attack on nonlinear filter generators," *IEEE Trans. Comput.*, vol. 49, no. 10, pp. 1100–1109, Oct. 2000.

[26] S. Chitravel and J. Ashta Lakshmi, "Linear complexity of second order PN_sequences addition with single order PN_sequence in nonlinear filter generator," *J. Discrete Math. Sci. Cryptogr.*, vol. 20, no. 5, pp. 1173–1181, Nov. 2017.

[27] T. Veugen. (2011). *Comparing Encrypted Data*. [Online]. Available: http://msp.ewi.tudelft.nl/ sites/default/files/Comparingencrypteddata.pdf

[28] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proc. 14th Annu. ACM Symp. Theory Comput. (STOC)*, 1982, pp. 365–377.

[29] I. Damgard, M. Geisler, and M. Kroigaard, "Efficient and secure comparison for on-line auctions," in *Proc. 12th Australas. Conf. Inf. Secur. Privacy*, vol. 4586, Jul. 2007, pp. 416–430, doi: 10.1007/978-3-540-73458-1_30.

[30] *UCI Machine Learning Repository*. Accessed: Jun. 14, 2021. [Online]. Available: https://archive.ics.uci.edu/ml/index.php

[31] *Weka Wiki Documentation*. Accessed: Jun. 10, 2021. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/documentation.html

[32] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, Cham, Switzerland: Springer, 2010, pp. 1–23.
[33] *How Many Primes are There?* Accessed: Jun. 10, 2021. [Online]. Available: https://primes.utm.edu/howmany.html#pnt
[34] *CS 255 (Introduction to Cryptography).* Accessed: Jun. 10, 2021. [Online]. Available: https://www.cs.virginia.edu/dwu4/notes/CS255LectureNotes.pdf
[35] X. Liu, R. Lu, J. Ma, L. Chen, and B. Qin, "Privacy-preserving patient-centric clinical decision support system on Naïve Bayesian classification," *IEEE J. Biomed. Health Informat.*, vol. 20, no. 2, pp. 655–668, Mar. 2016, doi: 10.1109/JBHI.2015.2407157.
[36] *The China Physiological Signal Challenge 2018: Automatic Identification of the Rhythm/Morphology Abnormalities in 12-Lead ECGs*. Accessed: Jun. 10, 2021. [Online]. Available: http://2018.icbeb.org/Challenge.html
[37] A. M. Koya and P. P. Deepthi, "Efficient on-site confirmatory testing for atrial fibrillation with derived 12-lead ECG in a wireless body area network," *J. Ambient Intell. Hum. Comput.*, Nov. 2021, doi: 10.1007/s12652-021-03543-9.
[38] *UCI Machine Learning Repository*. Accessed: Jun. 14, 2021. [Online]. Available: https://archive.ics.uci.edu/ml/index.php
[39] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proc. 14th Annu. ACM Symp. Theory Comput. (STOC)*, 1982, pp. 365–377.
[40] D. Bortolotti, M. Mangia, A. Bartolini, R. Rovatti, G. Setti, and L. Benini, "Energy-aware bio-signal compressed sensing reconstruction on the WBSN-gateway," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 3, pp. 370–381, Jul. 2018.

SONA ALEX (Member, IEEE) received the B.Tech. degree in electronics and communication engineering from the University of Calicut, Kozhikode, India, in 2007, and the M.Tech. degree in VLSI and embedded systems from the Cochin University of Science and Technology, Kochi, India, in 2011. She is currently pursuing the Ph.D. degree with the Department of Electronics and Communication Engineering, National Institute of Technology Calicut, Kozhikode. Her research interests include m-healthcare systems, cloud computing, distributed systems, and cryptographic system implementations.

K. J. DHANARAJ (Member, IEEE) received the B.Tech. degree in electronics and communication engineering and the M.Tech. degree in digital systems and communication engineering from the National Institute of Technology Calicut, in 2003 and 2007, respectively, and the Ph.D. degree from the Indian Institute of Technology Delhi, New Delhi, India, in 2017. He is currently working as an Assistant Professor at the National Institute of Technology Calicut. His research interests include power management IC design, low power analog/digital IC design for neuromorphic and neural network chips, architectures for signal processing, and cryptographic systems.

P. P. DEEPTHI (Senior Member, IEEE) received the B.Tech. degree in electronics and communication engineering from the N. S. S. College of Engineering, Palakkad (University of Calicut), in 1991, the M.Tech. degree in instrumentation from the Indian Institute of Science, Bengaluru, in 1997, and the Ph.D. degree in secure communication from the National Institute of Technology Calicut, in 2009. She has been working as a Faculty in institutions under IHRD, Thiruvananthapuram, from 1992 to 2001, and the Department of Electronics and Communication Engineering, National Institute of Technology Calicut, since 2001. Her current research interests include signal processing with security applications, cryptographic system implementations, information theory, and coding theory.