SAFE: SDN-Assisted Framework for Edge–Cloud Interplay in Secure Healthcare Ecosystem

Gagangeet Singh Aujla, *Member, IEEE*, Rajat Chaudhary, *Student Member, IEEE*, Kuljeet Kaur, *Student Member, IEEE*, Sahil Garg, *Student Member, IEEE*, Neeraj Kumar, *Senior Member, IEEE*, and Rajiv Ranjan, *Senior Member, IEEE*

***Abstract*—Improved quality of life has lead the healthcare industry to geographically expand and support real-time services. Following this trend, a surge of healthcare monitoring devices has substantially overgrown in the global market. These devices tend to generate data in humongous quantity that need real-time analysis with seamless and secure transmission to the computing nodes. The existing computing and networking infrastructures fall short to cater the services with desirable quality of service. Hence, to overcome these challenges, the proposed work presents a comprehensive platform referred as software defined network (*SDN) Assisted Framework for Edge–Cloud Interplay* in Secure Healthcare Ecosystem (*SAFE*). The objectives of SAFE include: first, an offloading scheme to support edge–cloud interplay, second, an SDN-assisted virtualized flow management scheme, and, third, a secure Lattice-based cryptosystem. Finally, the proposed scheme is validated on different performance parameters. Additionally, a security evaluation of the designed cryptosystem is also presented. The results obtained indicate the supremacy of the designed framework.**

***Index Terms*—Edge–cloud interplay, healthcare ecosystem, lattice-based cryptosystem, software-defined networks (SDN).**

I. Introduction

With the progressive improvements now being made in living standards, the consciousness amongst the masses for good health assurance has been substantially enhanced. As a virtue of this fundamental transformation, the number of healthcare monitoring devices available in the market has witnessed a massive propulsion. These devices include a wide variety of biosensors, smart wearable gadgets (such as smart phones, smart wrist watches, smart bracelets, and smart clothes), motion sensors, etc. This gradual shift in the healthcare industry has led to distributed and patient-based approaches from the traditional centralized and disease specific approaches [1]. This is mutually endorsed by the acceptance of Internet of things (IoT) in the healthcare industry, which can be attributed to the emergence of cyber physical systems under Healthcare 4.0. However, with their rapid acceptance in the global market, the amount of the healthcare data generated every second has exponentially grown in terms of volume, veracity, variety, and velocity. This is even evident from the statistics shared by IDC in the year 2015 [2]. According to IDC, healthcare industry data are projected to increase upto 2k exabytes by the year 2020. In general, it can be concluded that the healthcare industry is the next big producer of big data in the coming years. This transfiguration is bound to impose further complexities in terms of real-time data analytics and transmission for providing real-time healthcare services.

The traditional networking infrastructures require every bit of data to be sent to the remote central cloud repository for further processing and analysis. However, these remote repositories are collocated at distant geographical locations; which fall short to provide adequate quality of service (QoS) and quality of experience (QoE) measures due to the involved network congestion and traffic. The related issues encapsulate greater data transmission latency, reduced response delay, limited data availability, and higher processing time, etc. [3]. This calls for the entire reconfiguration of the underlying healthcare infrastructure for furnishing the designated services with high QoS and QoE.

In order to cater above-mentioned challenges, Cisco recently came up with the novel concept of edge computing: An extension to the centralized cloud computing infrastructure. The former supports the deployment of virtualized computing platform by bringing the computation intensive tasks closer to the user. This is achieved by deploying the computing nodes (such as micro DCs, nano DCs, and laptops) at the edge of the network. In contrast to the cloud, edge nodes are more geographically

Manuscript received November 30, 2017; revised April 29, 2018; accepted August 15, 2018. Date of publication August 24, 2018; date of current version January 3, 2019. Paper no. TII-17-2856. *(Corresponding author: Neeraj Kumar.)*

G. S. Aujla is with the Computer Science and Engineering Department, Chandigarh University, Mohali 140413, India (e-mail: gagi_aujla82@yahoo.com).

R. Chaudhary, K. Kaur, S. Garg, and N. Kumar are with the Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala 147004, India (e-mail: rajatlibran@gmail.com; kuljeet0389@gmail.com; garg.sahil1990@gmail.com; neeraj.kumar@thapar.edu).

R. Ranjan is with the School of Computer, Chinese University of Geosciences, Wuhan 430074, China, and also with the School of Computing, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K. (e-mail: raj.ranjan@ncl.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2018.2866917

dispersed and have the ability to cater users requests on real-time basis. This notion was further asserted by Wang and Yi [4], which clearly proved the efficacy of the edge computing infrastructure to support large-scale mobile services with reduced latency and bandwidth burden. Furthermore, it imparts single hop communication to the mobile users and associates with the cloud to access its high-end functionalities [5]. So, edge computing is considered as a powerful extension to the cloud rather as an alternative to it. Due to this, the interplay between the cloud and edge is considered as an important research direction, and it has been exploited by a number of researchers to address various issues ranging from communication delay to energy efficiency [6], [7].

In light of the above-mentioned advantages of edgecloud interplay, the proposed work models the healthcare ecosystem as the hierarchy of edge and cloud computing nodes across different geographically dispersed clinical centers (CCs). However, the major challenges in the considered setup involve the transmission of every bit of data either to the central cloud or scattered edge computing nodes; for real-time data storage, mining, and analysis. However, this bulk data transmission may lead to network congestion and throttling issues across the communication channel. Hence, it is essential to reconfigure the existing healthcare system with competent technologies that can make well-informed decisions about traffic rerouting to the available computing nodes. This is attainable by furnishing better data transmission competence using 5G-enabled communication technology along with the software-defined network (SDN) infrastructures.

The concept of SDN was first introduced by Stanford University's research team SLATE and has evolved as the next big thing in the domain of communication and networking domain. In contrast to the traditional networking infrastructures, the modern SDN has the competence to virtualize the networking infrastructures [8]. This is primarily attained by separating the control plane from the data plane. Furthermore, SDN's network virtualization comes with several advantages, such as effortless and dynamic network spanning, and tweaking of underlying networks to cater different applications. Due to these reasons, SDN-based networking infrastructure has been adopted in the proposed work for providing efficient healthcare services on real-time basis. This is primarily achieved by unleashing the dynamic data offloading policies between the hierarchy of edge devices and cloud computing servers. The smart controller employed in the SDN-based networking infrastructure helps to either reroute the network traffic toward the cloud computing servers from the edge nodes (forward offloading) or from cloud to the edge nodes (reverse offloading), respectively. This logic is based on the availability of computing and networking resources on the designated computing nodes.

The overall strategy illustrated previously plays a vital role in enhancing the overall QoS and QoE level of the healthcare services in terms of reduced computational cost and network latency. However, a significant challenge in front of the proposed healthcare ecosystem would be data security and privacy. According to IDC [2], merely 57% of healthcare data are convincingly protected today and almost 93% of data requires robust protection. Hence, designing of secure and robust cryptosystem is inevitable for future healthcare applications [1], [9]–[11].

A. Motivation

According to [12], SDN's smart and secure networking infrastructure along with 5G would play a significant role in making health services a "disruptive technology." In addition to this, the interplay between the edge and cloud is another significant infrastructural demand of the future real-time operations and applications; particularly for the Health 4.0. Analyzing the powerful presence of 5G, SDN, edge, and cloud in the umbrella of IoT domain, their presence in the future healthcare ecosystems is undoubtedly unavoidable. Thus, the primary agenda of the proposed work is to integrate these technologies and built a comprehensive platform, i.e., *SDN-Assisted Framework for Edge–Cloud Interplay in Secure Healthcare Environment (SAFE)*.

B. Contribution

To mitigate the above-mentioned issues, following contributions are presented in this paper.

1) An efficient edge–cloud interplay for healthcare ecosystem has been designed by the leveraging the benefits of SDN for forward and reverse data offloading.
2) Furthermore, an SDN-assisted virtualized flow management strategy across multiregion has been formulated.
3) A secure Lattice based cryptosystem for encryption/decryption and authentication has been modeled and validated on different evaluation criteria.

C. Organization of the Paper

The rest of this paper is organized in the following sequence. Section II illustrates the system model of the proposed SAFE framework. The detailed technical description of the proposed scheme is presented in Section III. Following this, observations and evaluation are presented in Section IV. Finally, the paper is concluded in Section V.

II. SDN-Based System Model

The various aspects related to the SDN-based system model are discussed in the subsequent sections.

A. Layered Healthcare Architecture

Fig. 1 shows the layered architecture of the smart healthcare system comprising hierarchy of CCs ranging from layer *0* (lowest layer) to layer *4* (topmost layer). Layer *0* consists of end user domain (patients, doctors, ambulance, wearable devices, and sensors). Biosensors, such as blood pressure, electrocardiogram, electromyography, insulin, and electroencephalography, sense the physiological parameters of patient that are either processed/analyzed or stored at remote locations. Layer *1* consists of rural CCs located in the vicinity of the end user for a group of rural establishments. Layer *2* comprises urban CCs followed by layer *3* dedicated to multiple regional CCs. In each region,

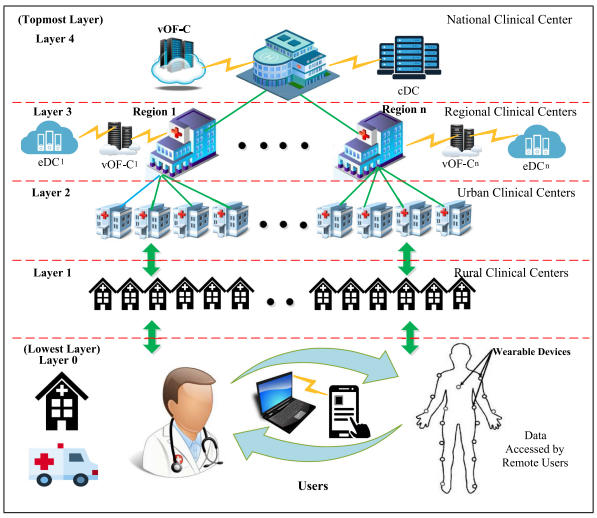

Fig. 1. System model of the proposed scheme.

a dedicated edge data center (eDC) is located to handle the activities of regional CCs and all the lower layers. Layer 4 is the topmost layer of the hierarchy that consists of national CC connected to a centralized cloud DC (cDC). The cDC or eDC are responsible for storing, collecting, and processing the data acquired from various lower layers.

B. SDN-Based Communication Model

In this paper, a SDN-based communication model is presented for healthcare ecosystem. Each layer is connected to the other using an open flow (OF) communication model. In this model, three distinct planes; 1) data, 2) control, and application planes are core components. These planes are elaborated as below.

Data plane: The forwarding devices (OFswitches, OFrouters, etc.) resides at this plane and follow the flow rules established by the OF controller to forward the traffic. The OFswitches abide to the flow entries (FEs) stored in their flow tables (FT). These FEs are set by the OF controller through a control algorithm. Multiple FTs are connected to each other using a pipeline [13]. SDN architecture is capable of creating virtual instances of physical switches (pS_i) known as virtual switches (vS_i).

Control plane: The control plane popularly known as brain of the SDN architecture, is the logically centralized decision making plane. In this plane, the logical brain, i.e., physical OF controller (pOF-C) resides. pOF-C works according to a reprogrammable control algorithm hosted by a centralized server. The major tasks of the pOF-C is to provide control decisions, commands, and instruction set that for smooth flow of data from the source host (H_{src}) to the destination host (H_{dst}).

Application plane: In this plane, different applications, such as virtualization, load balancing, flow scheduling, and fault tolerance are deployed.

C. Problem Formulation

The major objective of the proposed model is to select an optimal eDC or cDC for offloading the data, service, or application in case of starvation of resources. In such as multi edge–cloud ecosystem, the main entities that are vital comprises; source DC (i), flow path (j), and destination DC (k). Now, for selecting optimal destination DC, multiple choices exists for offloading from ith DC to kth DC with respect to j flow paths. For this purpose, following mapping ($\hat{\upsilon}_{i,j,k}$) exists:

$$\hat{\upsilon}_{i,j,k} = \sum_{i=1}^{n} \begin{bmatrix} 1,1,1 & 1,2,1 & . & . & 1,j,1 \\ 1,1,2 & 1,2,2 & . & . & 1,j,2 \\ . & . & . & . & . \\ . & . & . & . & . \\ 1,1,k & 1,2,k & . & . & 1,j,k \end{bmatrix}. \quad (1)$$

Now, to select optimal ijk pair, a combined utility function is defined as follows:

$$\upsilon_{ijk} = \frac{\beta_{rq} \times \theta_i^{av}}{(n+1) \times \tau_i^{av}} \times \frac{1}{d_{\frac{i \to k}{j}}} \quad (2)$$

where $\beta_{rq}, \theta_i^{av}, \tau_i^{av}$, and $d_{\frac{i \to k}{j}}$ represent the required bandwidth, average anticipated throughput, and delay of the network after including the new load, and the distance from ith DC to kth DC through jth flow path.

Now, a decision variable ($\psi_{ijk}, \forall t$) to select the optimal ijk pair from the previously discussed matrix is defined as follows:

$$\psi_{ijk} = \begin{cases} 1 & \text{for} \quad \upsilon_{ijk} > \upsilon^*_{ijk} \\ 0 & \text{for} \quad \text{otherwise} \end{cases} \quad (3)$$

where ijk^* represents all pairs other than ijk.

Therefore, the objective function of the proposed scheme is formulated as follows:

$$\max \left[\sum_{j=1}^{n} (\upsilon_{1j_11})\psi_{1j_11} + \upsilon_{1j_22}\psi_{1j_22} + \cdots + \upsilon_{1j_nk}\psi_{1j_nk} \right] \quad (4)$$

subject to following constraints:

$$\psi_{ijk} \in [0,1] \quad (5)$$

$$\upsilon_i(k) > \upsilon_i(k^*) \quad (6)$$

$$\upsilon_k(t) > \upsilon_k(t-1) \quad (7)$$

$$d_{(\frac{i \to k}{j})} < d_{(\frac{i \to k}{j})^*} \quad (8)$$

where $U_i(k)$ is the utility of ith DC with respect to kth DC, $U_i(k^*)$ is the utility of ith DC with respect to DCs other than k, $U_k(t)$, and $U_k(t+1)$ are utilities of kth DC at time t and $t+1$, respectively, and $d_{(\frac{i \to k}{j})^*}$ denotes distance between all pairs other than ith to kth DC through flow path j.

III. SAFE: Proposed Scheme

Three phases of SAFE are discussed in subsequent sections.

A. Offloading Scheme for Edge–Cloud Interplay

In this section, an offloading scheme for edge–cloud interplay is presented for healthcare ecosystem. This scheme involves two phases; 1) forward offloading, and 2) reverse offloading. In forward offloading, two case exist; 1) eDC to cDC and 2) eDC to eDC offloading. In reverse offloading, only one case, i.e., cDC

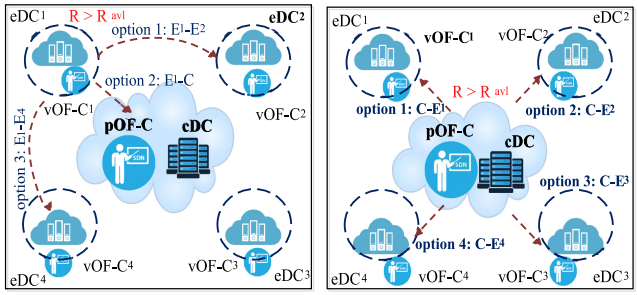

Fig. 2. Two-stage offloading scheme. (a) Forward offloading. (b) Reverse offloading.

TABLE I
CONDITIONS FOR INTER-DC MIGRATION

Case No.	Decision	Network resources	Computing resources
Case 1	True	✓	✓
Case 2	True (*)	x	✓
Case 3	False	✓	x
Case 4	False	x	x

(*) True for virtual network resources.

to eDC exists. Fig. 2 depicts both the phases of offloading process. Now, in order to participate in the offloading process, the conditions mentioned in Table I must be satisfied. The working of the offloading scheme is game inspired where i and k are considered as two players who make their decisions on the basis of the profits they receive. Therefore, separate profit functions are formulated for both the players. The profit function of ith eDC/cDC that send a request to all the available k cDC/eDCs is defined as follows:

$$\upsilon_i = a_i \ln(b_i + \Re) \tag{9}$$

where a_i and b_i are constants, ln function is used for preference ordering, and $\Re$ represents resource required.

Similarly, the profit function of kth eDC/cDC where offloading may take place is formulated as follows:

$$\upsilon_k = P_{\Re} \sum_{i=1}^{n} \Re \tag{10}$$

where $P_{\Re}$ price function for $\Re$.

Fig. 2 shows the two-way offloading process. It shows that if $\Re$ is more than $\Re_{\text{avl}}$, then the offloading takes place. Now, there may be multiple options for offloading. However, an optimal destination is required for the offloading to take place. For this purpose, Algorithm 1 is designed. This algorithm works in two stages. In first stage, i eDCs initiates *FORWARD-OFFLOADING* procedure by announcing the resources required $(\Re : \alpha_{tp}, \beta_{rq}, \varsigma_{rq})$ to k (cDC or eDCs). Here, $\Re$ comprises application type (α_{tp}), bandwidth requirement (β_{rq}), and computing resources (ς_{rq}) (line 1–5). Now, each available eDC or cDC checks $\Re$ with the $\Re_{\text{avl}}$ with them. If $\Re$ is available, then $\upsilon_k(t)$ is computed using (9). If the value of $\upsilon_k(t)$ is higher than its value in the previous time-slot, then all the conditions in Table I are *true*. After this, $\upsilon_i(k)$ is computed using (10). If the value of

Algorithm 1: Offloading Process.

Input: i ($eDCs$ or cDC), k ($eDCs$ or cDC), j (flow path)
Output: ijk pair

```
1:  procedure FunctionFORWARD-OFFLOADING
2:      for (i = 1; i ≤ n; i++) do
3:          Check ℜ : (α_tp, β_rq, ς_rq) ▷ ℜ: required resources
4:          for (k = 1; k ≤ n; k++) do
5:              ℜ → k        ▷ Announce ℜ to available eDCs or cDC
6:              ℜ → ℜ_avl
7:              if ℜ is available then
8:                  Compute υ_k
9:                  if υ_k(t) > υ_k(t − 1) then
10:                     TRUE
11:                 end if
12:             end if
13:             Compute υ_i(k)
14:             if υ_i(k) > υ_i(k*) then
15:                 Add k in queue ϱ
16:             end if
17:             j ← Call Algorithm 2
18:             Map ijk pairs
19:             Compute υ_ijk
20:             if (υ_ijk > υ*_ijk) then
21:                 Set ψ_ijk == 1
22:                 Select ijk pair
23:                 Offload to selected k
24:             else
25:                 Select next pair and offload
26:             end if
27:         end for
28:     end for
29: end procedure
30: procedure Function(REVERSE-OFFLOADING)
31:     for (k = 1; k ≤ n; k++) do
32:         Repeat step 3-28
33:     end for
34: end procedure
```

$\upsilon_i(k)$ is higher than the value of $\upsilon_i(k^*)$, where k^* is the set of all other eDCs/cDCs other than k. Now, add such eDCs or cDC in queue ϱ (line 6–16). Once the destination queue is ready, the flow path (j) is computed using Algorithm 2 for each element in ϱ (line 17). Using j, map all ijk pairs. For all ijk pairs, compute υ_{ijk} using (2) (line 18 and 19). Now, υ_{ijk} is compared with υ^*_{ijk}, ($^*ijk$ is the set of all ijk pairs other than ijk). If υ_{ijk} is more than υ^*_{ijk}, then pair ijk is selected. Otherwise, the next pair is selected. Finally, the offloading is performed using the selected ijk pair (line 20–29). Similarly, the procedure for *REVERSE-OFFLOADING* is performed. In this case, cDC become i and the step 3–18 are repeated to select optimal ijk pair (line 30–34).

B. Multiregion Virtual Flow Management Scheme

The standard OF model supports a single centralized controller based architecture. However, such architecture suffers

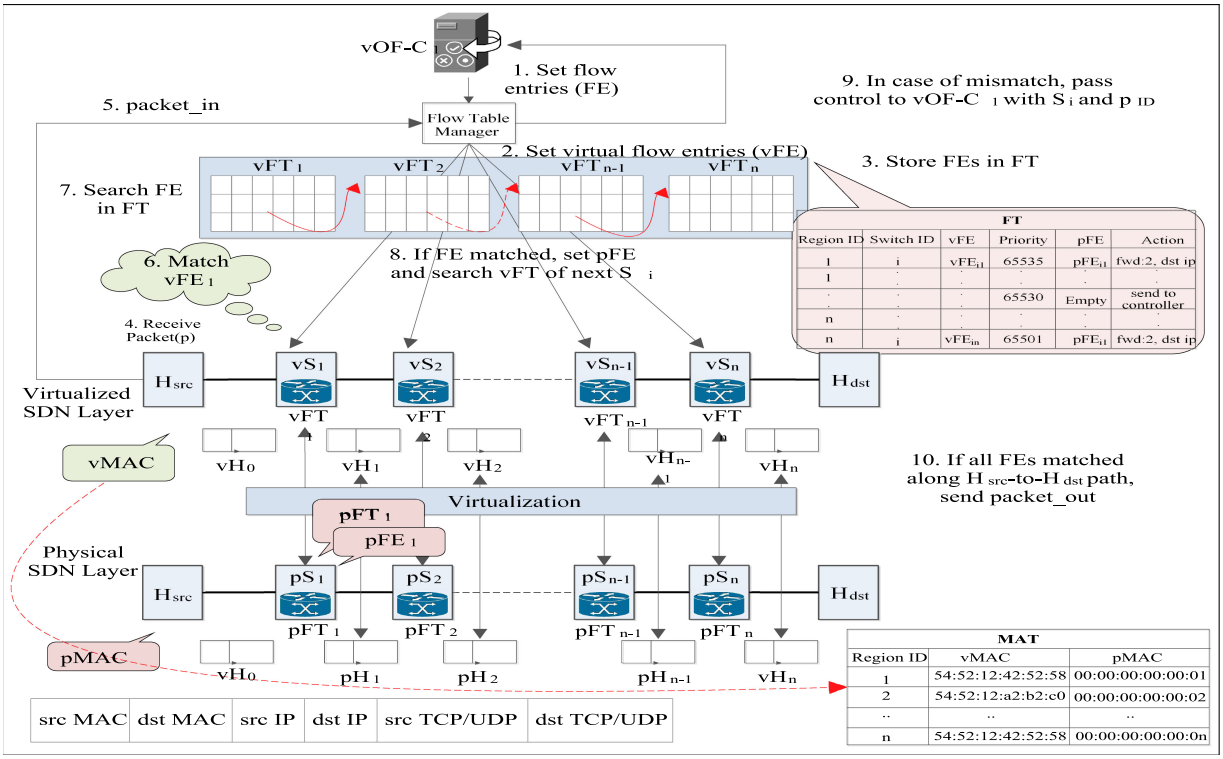

Fig. 3. Working of control flow scheme.

from several bottlenecks, such as fault tolerance, resilience, and inefficient resource utilization. Moreover, in healthcare ecosystem, the sharing of network infrastructure among parallel applications is utmost important to reduce the overall expenses. Therefore, to overcome these issues, in this paper, a multiregion virtualized OF architecture is presented for the healthcare system. In this architecture, pOF-C is deployed in a single physical OF network of the healthcare ecosystem. The proposed multiregion architecture contains virtual SDN layer comprising logical entities corresponding to the physical SDN layer comprising physical entities. It contains one or more regions (layer 3) comprising a dedicated virtual OF controller (vOF-C) and a group of end hosts (in lower layers). These vOF-Cs are created over pOF-C and deployed in each region. The vOF-Cs are responsible to manage the subsequent layers and devices that are part of the region.

Each region is isolated from each other and use their respective vOF-C as an exclusive network. For example, in a network comprising two regions (1 and 2), two virtual controller (vOF-C_1 and vOF-C_2) are created such that both are distinct and fulfill the condition: vOF-$C_1 \neq$ vOF-C_2. In this way, vOF-Cs can implement their own flow controls to run their applications over a single pOF-C concurrently. The communication between different regions occur through the global controller (pOF-C) in the proposed scheme. This has been done to avoid the complexity and congestion of intercontroller administrative traffic with the routine data traffic. However, in future, an efficient mechanism could be devised for the direct communication between different regions.

TABLE II
FLOW TABLE (FT)

R_{ID}	S_i	**vFE**	**Priority**	**pFE**	Action
1	1	vFE_{11}	65535	pFE_{11}	fwd, $pMAC_i$
..	..	..	..	..	..
..	..	..	..	..	..
n	p	vFE_{np}	0	pFE_{11}	$pMAC_p$

The global pOF-C sets the FEs in the FTs of each pS. Table II depicts a typical FT with FEs, such as region id, switch id, vFE, priority, pFE, and action. Each FTs contains of a matching field (M_{pq}) and an action field. The M_{pq} of each FT contains the ingress port and header values. Now, when a data packet (p) travels from H_{src} to H_{dst}, then pS_i search for matching FE in the FT. Once a matching FE is found in the FT, then pS_i performs the corresponding action (modifying the header values). For example, when S_1 receives p with header value (H_0), it forwards it to S_1 by incrementing the header value to H_1 ($H_0 \neq H_1$). This process continues until p reaches H_{dst}. Each entry in the FT set by vOF-C is treated as a vFE and it corresponds to a pFE at pS. In region *1*, a vFE^1_{pq} contains vM^1_{pq} and the pFE^a_{pq} contains pM^1_{pq}. Similarly, a virtual header (vH_i) corresponding to a physical header is denoted by pH_i also exists. To achieve network isolation for each region, the header value of each region is distinct and private. For example, in regions *1* and *2*, the vM^1_{pq} and vM^2_{pq} corresponding to pM^1_{pq} and pM^2_{pq} are unequal if $a \neq b$. Similar condition applies for pH^1_i and pH^2_i such that $1 \neq 1$. Fig. 3 shows the mapping of virtual and physical SDN layers.

TABLE III
MAC ADDRESS TRANSLATION TABLE (MAT)

R_{ID}	vMAC	pMAC
1	54:52:12:42:52:58	00:00:00:00:00:01
..	..	..
n	54:52:12:a2:b2:c0	00:00:00:00:00:0n

The proposed scheme supports translation of message authentication code (MAC) address in header and MF_{pq} of p unlike other multitenant OF architectures. The major reasons are as follows.

1) Isolation with pMAC address is possible.
2) Applicable to all Ethernet types.
3) The MAC address space length (48 b) is enough allocate pMAC to all vMACs of each region.

For MAC address translation, initially the values of pMAC address for virtual pairs (vMAC and R_{ID}) are generated. Table III shows the MAC address translation table (MAT). Initially, when the translator send a query, the MAT is checked for pMAC corresponding to vMAC and R_{ID}. If the address exists in the MAT, then the pMAC is returned. But, if pMAC does not exists in MAT, then the MAC address manager generates a new entry for pMAC address and register it in MAT for the corresponding virtual pair. Now, this new pMAC address is sent to the translator. The same process is followed when the translator needs a vMAC address and R_{ID} for a pMAC address. The complete working of the proposed scheme is shown in Fig. 3. The complete journey of p from H_{src} to H_{dst} in a step-by-step (step 1–10) illustration is presented. Moreover, Algorithm 2 (MRFMA) is presented to depict the flow of the proposed scheme.

C. Lattice-Based Cryptosystem for Healthcare

To protect the insecure channel from various security attacks (distributed denial of service, replay, and perfect forward secrecy) authentication plays a vital role. Similarly, an adversary may also launch various attacks (man-in-the-middle, active eavesdropping, known plaintext, and chosen ciphertext) on the information itself. In the healthcare ecosystem, each CC is connected through an SDN-based two-way communication system via a wired/wireless network. Therefore, the information needs to be kept confidential through secure encryption and decryption process. For the above-mentioned reasons, in this paper, a Lattice-based cryptosystem for healthcare ecosystem is designed that works in two phases; 1) lattice-based authentication scheme over Ring-LWE and 2) lattice-based data encryption scheme over ring-LWE. Table IV shows the list of notations used in the proposed cryptosystem.

1) Why Lattices?: The traditional public key cryptography (PKC) is primarily implemented by using the algorithms, such as RSA, Diffie–Hellman (DH) key exchange, elliptic curve cryptography (ECC), and finite field. However, all these cryptographic schemes are practically infeasible against quantum attacks. The PKC, such as RSA, DH key exchange are relatively slow for the voluminous amount of data. Moreover, some hard problems such as RSA is based on large integer factorization and ECC is based on the discrete logarithm problem;

Algorithm 2: MRFMA.

Input: $R_ID, S_i, p, vFE_i, vH_i, H_{src}$, Input port:$IP$
Output: pFE_i, Ouput port:OP

for $(R_{ID}, vOF - C)$ **do**
  Set *vFE* $\longleftarrow$ *pFE*
  Store *vFE* in *FT*
end for
for (R_{ID}, p, S_i) **do**
  if (*IP* is edge port) **then**
    Match *vFE*
    if (If *vFE* exists) **then**
      Set corresponding pFE
      if (Match $==$ exact) **then**
        for $(R_{ID}, vMAC)$ **do**
          if $R_{ID}, vMAC$ in *MAT* **then**
            Return *pMAC*
          else
            Create new *pMAC*
            Register new *pMAC* in *MAT* with $R_{ID}, vMAC$
            Return new *pMAC*
          end if
          Replace source *MAC* in *pM* by *vMAC*
        end for
        Set *pFE*
      else
        Match is wildcard
        Set *pFE* with higher priority
      end if
      pFE
      if (If *pFE* exists) **then**
        Set *pFE*
        Set output port *OP*
        if (*OP* is an edge port) **then**
          Add an action to modify source *MAC* address to *pMAC*
          $packet - out$
        else
          Replace source *MAC* in *pM* by *vMAC*
          Forward *p* to *IP* of S_{1+1}
        end if
      else
        Send $R_{ID}, p_{ID} \longleftarrow$ *vOF-C*
        Repeat step 1 to 26
        pFE matches
      end if
    else
      Send $R_{ID}, p_{ID} \longleftarrow$ *vOF-C*
      Repeat step 3 to 26
      pFE matches
    end if
  end if
end for

TABLE IV
NOTATIONS

Notations	Description
N	Prime integer modulus
P	Positive integer plaintext with index power of 2
l	Length of the bit string ($P \in \{0,1\}^l$)
R_N	Quotient ring ($R/_N R$)
$\phi_P(X)$	P^{th} Cyclotomic polynomial
X	Independent vectors of a group ${Z_N}^l$
$C, (c_1, c_2)$	Ciphertext
$d, (d_1, d_2, d_A, d_B)$	Private key vectors ($d \in {Z_N}^l$)
$e, (e_A, e_B)$	Public key vectors
$k, (k_1, k_2, k_A, k_B)$	Error vectors
g	Generator of cyclic group G such that ($g \in G$)
$\lfloor \rceil_2$	Modular rounding function to each coefficient of the polynomial
$\langle \rangle_2$	Cross rounding function
χ	Error distribution over R_N
$random$	Randomized function
rec	Reconciliation function
h	Secure one-way hash function
H	Hint function
E_r	Robust extractor
s_A, s_B	Session key
$D_{Z^l,\beta N}$	Discrete Gaussian distribution
β	Error parameters (real number $\in \{0,1\}$)
T	Transpose
U	Uniform random matrix of $n \times n$ order ($U \in {Z_N}^{l \times l}$)
σ	Gaussian parameters ($\sigma = 8/\sqrt{2\pi}$)
ϕ	Euler quotient function
ID_A, ID_B	Identity of device A and device B
$V, (V_A, V_B)$	Random vector
$\|\|$	Concatenation

Fig. 4. Secure authentication based on Ring-LWE.

therefore, it becomes difficult to factorize the large numbers or problematic to compute discrete logarithms in a finite group. The modern lattice based cryptography is based on quantum cryptography is believed to be resistant against quantum attacks and utilizes the laws of quantum mechanics, number theory, and algebra. Furthermore, lattice is represented in matrix form for efficient storage utilization and fast Fourier transform and modulo (mod) function is used for executing faster matrix arithmetic operations. Lattice cryptography is resistant against quantum attacks and breaking the security is equivalent to solving NP-hard problems. Therefore, it is beneficial from security point of view that lattice-based cryptography is selected in the proposed ecosystem.

2) Basis of Lattices: The elementary elements of abstract algebra in modern cryptography consists of group, ring, and field. The difference between the three terms is based on their mathematical properties. A group is defined as $\{G, \cdot\}$ where $(\cdot)$ implies a binary operator like addition or multiplication of elements in a set. With the help of a binary operator, four properties, such as closure, associative, identity, and inverse of elements can be followed. A superset of a group is a cyclic group that follows an extra properties of commutative of addition and exponentiation within a group. The second component is a ring defined by $\{R, +, \times\}$ where; addition and multiplication operation can be used simultaneously. A ring follows an additional properties, such as closure under multiplication, associative property of multiplication, and distributive. The third component is field defined by $\{F, +, \cdot\}$ and is a superset of ring with an additional property of multiplicative of inverse. Let us assume Z is defined as a ring of integers and Z^l be a set of integer coordinates defined over the field R^l in l dimensional vector space.

The lattice-based cryptography is defined over the ring and field of abstract algebra. The term lattice is defined as a regular ordered arrangement of isolated vectors in two-dimensional space or the set of all integer linear combination of l linearly independent vectors. A lattice is defined over the ring of vector modulo to perform faster matrix arithmetic operations in high-dimensional space. The algebraic structure of lattice is represented over the cyclotomic ring as: $R = \frac{Z_N[X]}{\Phi_P(X)} = R = \frac{Z_N[X]}{(X^l+1)}$ where $(X^l + 1)$ is a irreducible polynomial equation of utmost degree $(l - 1)$ [14]. The polynomial equation is defined as follows:

$$(a_0 v + a_1 v x^1 + a_2 v x^2 + \cdots + a_{l-1} v x^{l-1})(\text{mod } N) = v(a_0 + a_1 x^1 + a_2 x^2 + \cdots + a_{l-1} x^{l-1})(\text{mod } N) \quad (11)$$

where v is a vector that forms a lattice basis and $(a_0, a_1, \ldots, a_{l-1})$ are the coefficients. Here, Z is the ring of rational integers and $\phi_P(X)$ is the cyclotomic polynomial.

3) Lattice-Based Authentication Scheme Over Ring-LWE: Lattice is a prominent technique used to provide resilience against quantum attacks. The traditional RSA and ECC algorithms are susceptible to quantum attacks and are inefficient for small devices with 8-bit microcontrollers. Therefore, a lattice-based authentication scheme over Ring-LWE is designed. The proposed authentication scheme is divided into three-phases comprising login, verification, and shared session key as shown in Fig. 4. These are as follows.

1) *Login phase:* The login phase initiates after the user (patient, doctor) completes the successful registration process. Hence, for every user, an account is created in the database. Now, when the user wants to access his/her account, the user first submits the login credentials. Suppose, a device A wants to retrieve any information from device B, then the steps followed by device A

are explained as follows: Device A enters his/her identity and password $(\mathrm{ID}_A, \mathrm{PW}_A)$ along with a random number (V_A). The user's private information is kept confidential by using a one-way hash function computed as: $(L_A) = h(\mathrm{ID}_A||\mathrm{PW}_A||V_A)$. Finally, the device A sends (ID_A, L_A) to the device B.

2) *Verification phase:* The device B verifies the login process by first choosing a private key as $(d_b \in {Z_N}^l)$, and $d_B \leftarrow D_{Z^l,\beta N}$ as a discrete Gaussian distribution [15]. The device B then computes $O_A = h(\mathrm{ID}_A||d_B), P_A = O_A \oplus L_A$, and verifies whether ${L_A}' = h(\mathrm{ID}_A||\mathrm{PW}_A||V_A)$ matches. If the verification fails, then the session terminates, else the login is considered successful. For a successful login, the device B sends the message to device A to send its public key.
3) *Shared session key phase:* The device A first choose a private key vector as (d_A) defined over the cyclotomic ring (R_N) based on the Ring-LWE method. Let χ be a probability distribution defined over R_N, then $d_A \xleftarrow{\$} \chi$ and error vector $(k_A \in Z^l), (k_A \xleftarrow{\$} \chi)$ denotes sampling of elements $(d_A, k_A) \in R_N$ according to χ [15]. The elements (d_A, k_A) are chosen from the discrete Gaussian distribution sampled as: $(d_A \leftarrow D_{Z^l,\beta N}), (k_A \leftarrow D_{Z^l,\beta N})$. Now, using the security parameters (N, P, χ, l, β), the device A computes its public key e_A. The e_A is computed by using device A's private key d_A, and a uniform random matrix $U \in {Z_N}^{l\times l}$ along with an error vector k_A and sends e_A to the device B. The device B first computes its public key e_B by using (d_B, U, k_B), then the device B generates a common session key s_B based on e_A. Moreover, a function $E_R \in R_N$ is added as a robust extractor in order to guarantee that the two parties extract the same information with respect to a hint function H and generates the output as a signal function $\sigma \in \{0,1\}^l$ as a Gaussian parameter [16]. Now, the device B sends (e_B, s_B) to the device A. The device A now computes its session key s_A and matches it with the s_B. Finally, after a successful match, the device A and B can exchange their information securely over a public channel.

4) Lattice-Based Data Encryption Scheme Over Ring-LWE: Our scheme exploits the lattice-based public key cryptosystem defined over the Ring-LWE method that produces compact ciphertext length as compared to other LWE schemes, thus, achieves improved bandwidth. The rec reconciliation function [17] is used that enables to achieve similar agreement at the receiver side in terms of error vector over the ring element R_N. Furthermore, rounding functions over mod 2, for example; $\lfloor\ \rceil_2$ modular rounding function and $\langle\ \rangle_2$ cross rounding function are used in order to drop less-significant bits for further reducing the ciphertext length to some extent [15], [17]. Fig. 5 shows the secure lattice-based encryption scheme defined over the Ring-LWE method with plaintext message as P and ciphertext as C is given in four phases: key setup phase, key generation phase, encryption phase, and decryption phase. The following algorithms are explained as follows.

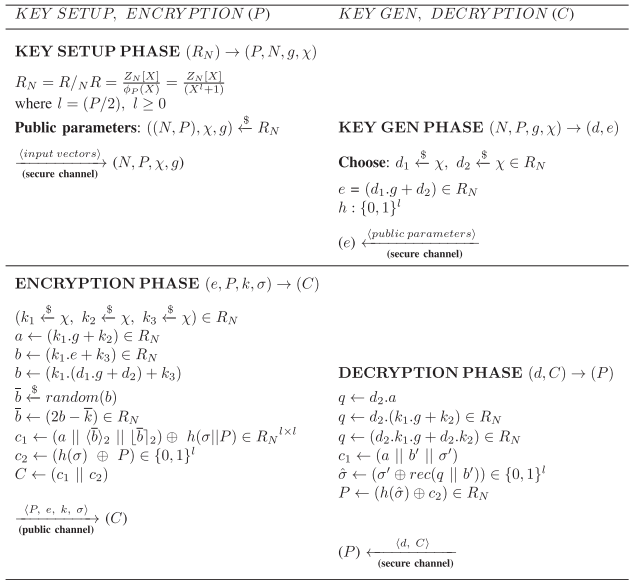

Fig. 5. Data encryption scheme using ring-LWE.

1) *Key setup phase:* The input parameters of the key setup process are defined as: g be the generator of cyclic group G such that $g \in G$ and $R_N = Z_N[X]/(X^l + 1)$ be the cyclotomic ring. The output produced in this phase includes the public parameters (N, P, χ, g), which are given as an input to the key generation phase.
2) *Key generation phase:* It takes the input as (N, P, χ, g) and returns the output as a private key vector d_1, d_2, and a public key vector e. The key generation phase samples error vectors k_1, k_2 from the χ in order to generate d_1, d_2, i.e., $(d_1, d_2) \leftarrow \mathrm{sample}(\chi) \in R_N$. Now, e is computed by using $d_1, d_2, g,$ i.e., $(d_1 \times g + d_2 \in R_N)$ and e is transmitted to the encryption phase over the secure channel.
3) *Encryption phase:* The inputs to this phase are the plaintext P, e, k, and a Gaussian parameters σ and it returns the output as C. The error vectors k_1, k_2, k_3 are sampled from χ, i.e., $k_1, k_2, k_3 \xleftarrow{\$} \chi$. The encryption phase first computes a, b by using e, g, and k_1, k_2, k_3. Now, a new element say $\overline{b} \in Z_N$ is sampled from the randomized function ran. The benefits of ran function is it avoids larger interval states by dividing the intervals of key stream into quadrant sets defined as: $\{(0, N/4), (N/4, N/2), (N/2, 3N/4), (3N/4, N)\} \in Z_N (\mathrm{mod}\ 4)$ where $\overline{b} = (2b - \overline{k})$ [17]. The key stream on which the two parties agree is produced by using $\lfloor\ \rceil_2$ as (mod 2), i.e., Z_2 or $(2/N \times b)(\mathrm{mod}\ 2)$ to every coefficients of $\overline{b}$ in order to round closer of 0 to $N/2$. Also, $\langle\ \rangle_2$ is applied over the $\overline{b}$ as a masking bit to provide sufficient information that a coefficient lies in which quadrant modulo N [17]. Now, the plaintext P is first concatenated with σ, then a hash value is computed and merged with $a, \lfloor b\rceil_2, \langle b\rangle_2$ to produce the ciphertext c_1. Finally, the ciphertext C is computed by concatenating c_1, c_2 and stored at the cloud.

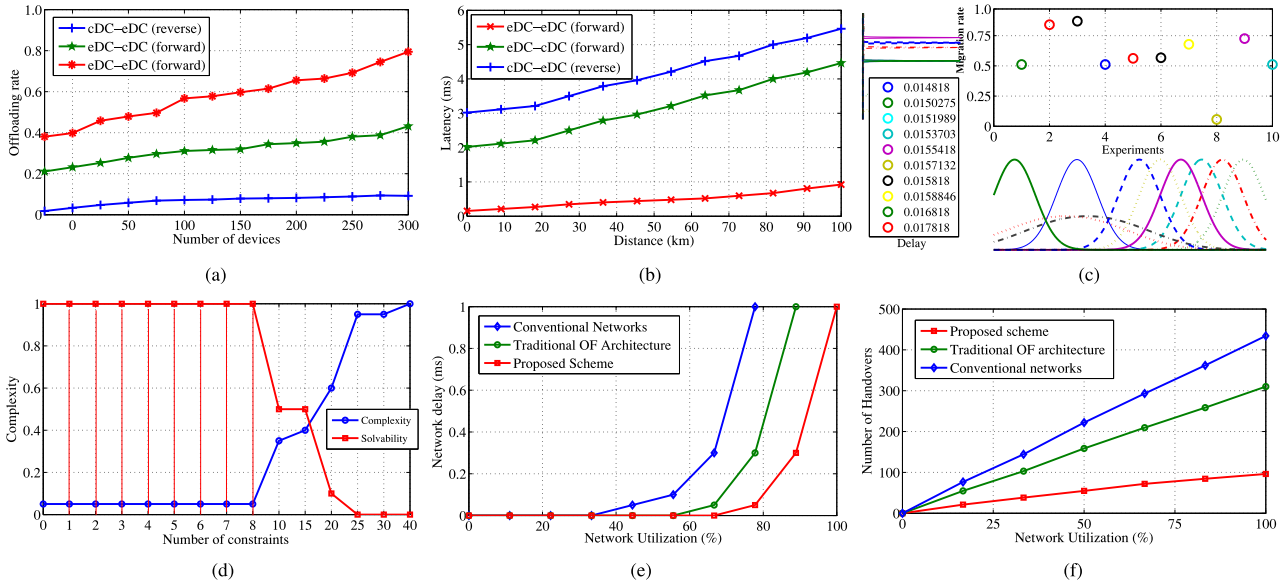

Fig. 6. Results obtained. (a) Offloading rate. (b) Latency. (c) Edge to edge analysis. (d) Complexity analysis. (e) Network delay. (f) Number of handovers.

4) *Decryption phase:* It takes input as (d, C) and computes the output as P. The rec function is used in order to generate the exact value at the decryption phase defined as $\text{rec} : Z_N \times Z_2 \rightarrow Z_2, R_N \times \{0,1\}^l \rightarrow R_2$. The ciphertext c_1 is verified correctly to secure it from the chosen ciphertext attack. Finally, the plaintext P is produced by $h(\hat{\sigma}) \oplus c_2$.

IV. Performance Evaluation

The proposed scheme is evaluated using simulated environment. The results and related observations are given in the following sections.

A. Simulation Results

First, SAFE is evaluated on the basis of offloading rate and latency. Fig. 6(a) shows the migration rate witnessed during the offloading process with respect to an increase in the number of healthcare devices. The eDC–eDC (forward) offloading shows highest migration rate. This is due to resource-constraint nature of eDCs. Moreover, the cDC–eDC (reverse) offloading shows lowest migration rate due to high resource availability at cDC. Fig. 6(b) shows the latency for various offloading scenarios with respect to increase in the distance. The eDC–eDC (forward) offloading witness lowest latency as compared to other offloading scenarios. The deep analysis of eDC–eDC is provided in Fig. 6(c). It shows the variation of migration rate with respect to delay for different experiments performed. Finally, the complexity analysis of the proposed scheme is provided. Fig. 6(d) shows the complexity variation and solvability analysis. It clearly shows that the optimization problem is easily solvable until eight constraints but after that its complexity increases drastically.

SAFE uses multiregion flow management scheme build over SDN architecture. Using this scheme, the optimal flow path is decided so as to reduce network delay with respect to an increase in the number of healthcare devices and distance between each device. The proposed scheme is compared with traditional SDN architecture and conventional networks. Fig. 6(e) shows the variation of network delay with respect to the utilization of network. SAFE uses a virtualized network architecture, thereby reducing the network delay with an increase in the network utilization. It is evident for the results obtained that the proposed scheme incurs lower delay as compared to other variant architectures. Similarly, the number of handovers that occur during the offloading process is also analyzed. Fig. 6(f) shows the number of handovers with respect to network utilization. The results depict that the proposed multiregion architecture involve lesser number of handovers as compared to other architecture.

B. Security Evaluation

The proposed scheme has been evaluated in terms of computation and communication costs as described below.

1) Computation Time: The computation time is computed for each phase of the encryption scheme is computed as follows.

Key setup and keyGen phase: Here, the operations used are two Gaussian sampling, one Fourier addition, and two Fourier forward operations. The average operation time taken by the Gaussian sampling is ≈0.265 ms, addition takes 1 ms, and fast Fourier forward takes ≈0.038 ms. Total operation time for a message p of size 1024 bits $\approx (2 \times 0.265 + 1 + 2 \times 0.038)$ ms ≈ 1.606 ms.

Encryption phase: Here, three Gaussian sampling, one random function, one $\langle \rangle_2$, one $\lfloor \rceil_2$, two addition, two hash function, and two $\oplus$ operations are used. The average operation time for random, $\langle \rangle_2$, and $\lfloor \rceil_2$ function ≈0.005 ms, while the Fourier multiplication takes ≈0.12 ms, the T_h operation takes ≈0.32 ms, and the $\oplus$ operation takes ≈0.0024 ms. Thus, the total operation time for $P = 1024$ bits $\approx (3 \times 0.265 +$

$3 \times 0.005 + 0.12 + 2 \times 1 + 2 \times 0.32 + 2 \times 0.0024)$ ms $\approx$ 3.5748 ms.

Decryption phase: The operations followed are one rec function, one Fourier backward, one multiply, one hash function, and two $\oplus$ operations. The average operation time of rec function takes $\approx$0.001 ms and the Fourier backward is $\approx$0.039 ms. Thus, the total operation time for $P = 1024$ bits $\approx (0.001 + 0.039 + 0.12 + 0.32 + 2 \times 0.0024)$ ms $\approx$ 0.4848 ms.

Finally, the overall execution time of all phases is $\approx (1.606 + 3.5748 + 0.4848)$ ms $\approx$ 5.6656 ms.

2) Communication Cost: Let us assume that P is 1024 bits, the identity is 128 bits, and the message digest (hash output) is 160 bits using SHA-1. The communication costs is computed for the authentication scheme for the device A and device B.

Device A: Initially, the bits processed by device A is L_A that takes the input bits, i.e., identity, password, random number as $(128 + 128 + 128) = 384$ bits and return the output as 160 bits message digest using SHA-1 and 128 bits identity. The message second transmitted by A is the public key (e_A), which is of 512 bits. Therefore, the communication bits processed by the device A is $\approx (128 + 160 + 512) = 800$ bits.

Device B: The message transmitted by the device B is the public key (e_B), and the hint function appended with the session key (σ). The communication bits required in processing (e_B, σ) is $(512 + 512) = 1024$ b.

So, total communication cost is $(800 + 1024) = 1824$ b.

C. Comparative Analysis

The proposed scheme has been compared with various existing proposals in two ways; 1) communication cost and computation time and 2) functionality and security features. The traditional PKC works on the multicore processor while the lattice cryptosystem is based on quantum computers. The silicon chip-based computer can store the information in binary representation as either 0 or 1 bit while the quantum computer operates on qubits (or quantum bits) represented in both 0 and 1 simultaneously. The core concept of quantum computer is based on the superposition principle in which a particle can exist in multiple states. First, in RSA algorithm, the key size of 2048-bits would require 4096 qubits to break while a ECC of 224-bits takes 1300 and 1600 qubits to break. Second, the time complexity of a multicore processor is exponential while the quantum computer takes polynomial time for a n-bit integer. Google 1000 qubits processor quantum computer represents $2^{1000} \approx 10^{31}$ operations concurrently. Google quantum computer is 100 million times faster than silicon-chip computer. Therefore, the PKC is vulnerable to active eavesdropping and this makes the PKC without the lattice-based quantum cryptosystem obsolescent instantly.

Table V shows the comparative analysis of the proposed scheme with various existing proposals on the basis of communication cost and computation time. The proposed scheme has been compared with the existing variants of its category in Table VI. The evaluation results clearly demonstrate that the proposed scheme performs better in comparison to the other existing schemes with respect to the known attacks in SG.

TABLE V
COMPARISON ANALYSIS

Scheme	Computation Time	CC	N_{ms}
[18]	$5T_{mp} + 2T_e + 12T_h + 2T_b \approx$ 505.72 ms	1920	3
[19]	$10T_{mp} + 2T_m + 5T_h + 2T_{E/D} + T_{cer} + T_{cver} \approx$ 532.43 ms	3648	4
[20]	$2T_h + T_{mac} + 1T_{hmac} + 2T_{E/D} \approx$ 12.48 ms	1696	3
[21]	$30T_h + 3T_{E/D} \approx$ 26.40 ms	4352	2
[22]	$20T_h + 3T_{E/D} \approx$ 23.20 ms	2272	2
[23]	$2T_h + 3T_e \approx$ 58.24 ms	4416	3
SAFE	$2T_{GS} + 1T_{FA} + 2T_{FF} + T_E + T_D \approx$ 5.665 ms	1824	4

CC: communication cost (bits), N_{ms}: number of messages, T_{mp}: multiplication point time, T_e: modular exponentiation time, T_h: one-way hashing time, T_b: bilinear pairing time, T_m : multiplication time, $T_{E/D}$: symmetric encryption and decryption time, T_{cer}: certificate time, T_{cver}: certificate verification time, T_{mac}: hashed MAC time, T_{hmac}: MAC time, T_{GS}: Gaussian sampling time, T_{FA}: Fourier addition time, T_{FF}: Fourier forward time, T_E: encryption time, T_D: decryption time.

TABLE VI
COMPARISON WITH EXISTING SCHEMES

Features	Gope *et al.* [24]	Bao *et al.* [25]	Bos *et al.* [15]	Lyubashevsky *et al.* [14]	Proposed
F1	✓	✓	✓	×	✓
F2	✓	✓	✓	×	✓
F3	×	✓	×	✓	✓
F4	✓	✓	✓	✓	✓
F5	✓	×	✓	×	✓
F6	×	×	✓	✓	✓
F7	×	✓	×	✓	✓
F8	×	–	×	✓	✓
F9	×	–	×	✓	✓

F1: mutual authentication; F2: perfect forward secrecy; F3: strong data encryption; F4: man in middle attack; F5: session key management; F6: resilient against distributed denial of service; F7: encrypted data storage; F8: known ciphertext attack; F9: known plaintext attack; ✓: represents that the particular scheme is secure; × : represents that particular scheme is insecure; –: not considered.

V. CONCLUSIONS

Healthcare industry has seen tremendous transformation in the last couple of years—from Healthcare 1.0 to Healthcare 4.0. However, these ever-changing technological shifts demand better computing and communicational infrastructures for high-end functionalities with QoS assurance. The said objectives have been achieved in the proposed work by integrating cloud and edge computing with SDN; to built SAFE. It is a composite framework designed especially for healthcare domain with the following three key contributions:

1) offloading scheme to support edge–cloud interplay;
2) an SDN-assisted virtualized flow management scheme; and
3) a secure Lattice-based cryptosystem.

The designed framework has been validated experimentally against the current state of the art techniques on the basis different performance metrics. The designed data offloading strategy has been verified on the basis of delay, complexity, and number of handovers. On the other hand, security evaluation of the designed cryptosystem has been achieved on the basis of computational and communicational cost. The results obtained clearly indicate the supremacy of the designed framework.

For a multiregion virtual flow management scheme, an optimal placement of virtual controller in different regions is one of the important issues that could be worked in future.

REFERENCES

[1] K. Zeb, K. Saleem, J. Al Muhtadi, and C. Thuemmler, "U-prove based security framework for mobile device authentication in ehealth networks," in *Proc. IEEE 18th Int. Conf. e-Health Netw., Appl. Serv.*, 2016, pp. 1–6.

[2] "IDC Predicts: Over 2K Exabytes of Healthcare Data by 2020, " 2015. [Online]. Available: https://www.calero.com/communications-lifecycle-management/blog-idc-predicts-over-2k-exabytes-of-healthcare-data-by-2020. Accessed on: Jul. 2017.

[3] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and M. Guizani, "Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 44–51, Feb. 2018.

[4] X. Wang and P. Yi, "Security framework for wireless communications in smart distribution grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 809–818, Dec. 2011.

[5] P. Borylo, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk, "Energy-aware fog and cloud interplay supported by wide area software defined networking," in *Proc. IEEE Int. Conf. Commun.*, May 2016, pp. 1–7.

[6] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay & power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.

[7] M. Aazam and E. N. Huh, "Dynamic resource provisioning through fog micro datacenter," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Mar 2015, pp. 105–110.

[8] G. S. Aujla, R. Chaudhary, N. Kumar, J. J. Rodrigues, and A. Vinel, "Data offloading in 5G-enabled software-defined vehicular networks: A Stackelberg-game-based approach," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 100–108, Aug. 2017.

[9] N. Kumar, K. Kaur, S. C. Misra, and R. Iqbal, "An intelligent RFID-enabled authentication scheme for healthcare applications in vehicular mobile cloud," *Peer-to-Peer Netw. Appl.*, vol. 9, no. 5, pp. 824–840, 2016.

[10] S. Alshehri, S. P. Radziszowski, and R. K. Raj, "Secure access for healthcare data in the cloud using ciphertext-policy attribute-based encryption," in *Proc. IEEE 28th Int. Conf. Data Eng. Workshops*, 2012, pp. 143–146.

[11] C. Hahn, H. Kwon, and J. Hur, "Efficient attribute-based secure data sharing with hidden policies and traceability in mobile health networks," *Mobile Inf. Syst.*, vol. 2016, 2016, doi: 10.1155/2016/6545873.

[12] C. Thuemmler, "IoT analytics for smart health and care." 2015. [Online]. Available: http://iot-week.eu/wp-content/uploads/2015/07/5-IoT-Analytics-for-smart-Health-and-Care-CThuemmler.pdf. Accessed on: Jun. 2017.

[13] G. S. Aujla, N. Kumar, A. Y. Zomaya, and R. Ranjan, "Optimal decision making for big data processing at edge-cloud environment: An SDN perspective," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 778–789, Feb. 2018.

[14] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *J. ACM*, vol. 60, no. 6, p. 43, 2013.

[15] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila, "Post-quantum key exchange for the TLS protocol from the ring learning with errors problem," in *Proc. IEEE Symp. Secur. Privacy*, 2015, pp. 553–570.

[16] J. Ding, X. Xie, and X. Lin, "A simple provably secure key exchange scheme based on the learning with errors problem," *IACR Cryptol. EPrint Arch.*, vol. 2012, pp. 1–15, 2012.

[17] C. Peikert, "Lattice cryptography for the internet," in *Proc. Int. Workshop Post-Quantum Cryptography*, 2014, pp. 197–219.

[18] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1900–1910, May 2018.

[19] D. Wu and C. Zhou, "Fault-tolerant and scalable key management for smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 375–381, Jun. 2011.

[20] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, "Lightweight and secure session-key establishment scheme in smart home environments," *IEEE Sensors J.*, vol. 16, no. 1, pp. 254–264, Jan. 2016.

[21] H. J. Kim and H. S. Kim, "Authhotp-hotp based authentication scheme over home network environment," in *Proc. Int. Conf. Comput. Sci. Appl.*, 2011, pp. 622–637.

[22] B. Vaidya, J. H. Park, S.-S. Yeo, and J. J. Rodrigues, "Robust one-time password authentication scheme using smart card for home network environment," *Comp. Commun.*, vol. 34, no. 3, pp. 326–336, 2011.

[23] F. K. Santoso and N. C. Vun, "Securing IOT for smart home system," in *Proc. IEEE Int. Symp. Consum. Electron.*, 2015, pp. 1–2.

[24] P. Gope and T. Hwang, "BSN-care: A secure IOT-based modern healthcare system using body sensor network," *IEEE Sensors J.*, vol. 16, no. 5, pp. 1368–1376, Mar. 2016.

[25] H. Bao and R. Lu, "Comment on "privacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 2–5, Feb. 2016.

Gagangeet Singh Aujla (S'15–M'18) received the B.Tech and M.tech. degrees in computer science and engineering from Punjab Technical University, Jalandhar, India, in 2003 and 2013, respectively, and the Ph.D. degree in Computer Science and Engineering from Thapar Institute of Engineering and Technology, Patiala, India, in 2018. For his Ph.D. work, he received IEEE TCSC Outstanding Ph.D. Dissertation Award (Award of Excellence) in 2018.

He is currently an Associate Professor with the Computer Science and Engineering Department, Chandigarh University, Mohali, India. He is also working as a researcher in India-Austria bilateral research project for cooperation in Science and technology at Thapar Institute of Engineering and Technology, Patiala, Punjab, India. He has many research contributions published in top cited journals, such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE COMMUNICATIONS MAGAZINE, the IEEE CONSUMER ELECTRONICS MAGAZINE, *Future Generation Computing Systems*, *Journal of Parallel and Distributed Computing*, *Computer Networks*, Information Sciences (Elsevier) and various International conferences such as IEEE GLOBECOM, IEEE ICC, IEEE WiMob, IEEE CCNC, ACM MOBIHOC, etc.

Dr. Aujla is member of the ACM.

Rajat Chaudhary (S'17) received the B.Tech degree in computer science and engineering from Uttar Pradesh Technical University, Lucknow, India, in 2010, and the M.Tech degree in Information Security Management from Uttarakhand Technical University, Dehradun, India, in 2012. He is working toward the Ph.D. degree in computer science and engineering from Thapar Institute of Engineering and Technology, Patiala, India.

He is currently a Junior Research Fellow with the Indo-Poland Joint project funded by Indian and Polish Governments. He has many research interests in the areas of computer networking, network security, and software-defined networks.

Kuljeet Kaur (S'15) received the B.Tech. degree in computer science and engineering from Punjab Technical University, Jalandhar, India, in 2011 and the M.E. degree in information security from Thapar Institute of Engineering and Technology, Patiala, India, in 2015, respectively; and the Ph.D. degree from Thapar Institute of Engineering and Technology, Patiala, India, in 2018.

Her main research interests include Cloud Computing, Energy Efficiency, Smart Grid, Frequency Support, and Vehicle-to-Grid.

Dr. Kaur has secured a number of research articles in top tier journals such as IEEE WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON SMART GRID, IEEE SENSORS JOURNAL, IEEE COMMUNICATIONS MAGAZINE, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE PS, Springer PPNA, and various International conferences such as IEEE GLOBECOM, IEEE ICC, IEEE CCNC, IEEE PES GM, ACM MOBICOM, etc.

Sahil Garg (S'16) received the B.Tech degree from Maharishi Markandeshwar University, Mullana, Ambala, India, in 2012, and the M.Tech degree from Punjab Technical University, Jalandhar, India, in 2014, both in computer science and engineering. He is working toward the Ph.D. degree in computer science and engineering from Thapar Institute of Engineering and Technology, Patiala, India. His research interests include machine learning, big data analytics, knowledge discovery, cloud computing, internet of things, and vehicular ad-hoc networks.

Some of his research articles have been published in top-tier journals, such as the IEEE NETWORK, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE COMMUNICATIONS MAGAZINE, the IEEE INTERNET OF THINGS JOURNAL, the IEEE CONSUMER ELECTRONICS MAGAZINE, *Future Generation Computing Systems*, Information Sciences (Elsevier), International Journal of Communication Systems (Wiley), and *Computers and Electrical Engineering* and and various International conferences of repute such as IEEE GLOBECOM, IEEE ICC, IEEE CCNC, ACM MOBICOM, etc.

Mr. Garg is member of the ACM.

Neeraj Kumar (M'15–SM'17) received the Ph.D. degree in CSE from Shri Mata Vaishno Devi University, Katra, India, in 2009

He was a Postdoctoral Research Fellow in Coventry University, Coventry, U.K. He is currently an Associate Professor with the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India. He has authored or coauthored more than 200 technical research papers in top cited journals and conferences. His research is supported by funding from UGC, DST, CSIR, and TCS.

Dr. Kumar is an Associate Technical Editor for the IEEE COMMUNICATION MAGAZINE. He is an Associate Editor for *International Journal of Computer Systems* (Wiley), *Journal of Network and Computer Applications* (Elsevier), and *Security and Communication* (Wiley).

Rajiv Ranjan (SM'15) received the B.E. degree in computer engineering from the North Gujarat University, Patan, Gujrat, India in 2002, and the Ph.D. from The University of Melbourne, Australia in 2009. He is Chair and Professor of computing science and internet of things with Newcastle University, Newcastle upon Tyne, U.K. He is an internationally established scientist with about 250 scientific publications and expertise in cloud computing, big data, and the Internet of things. He is an innovator with strong and sustained academic and industrial impact and a globally recognized R&D leader with a proven track record.

Prof. Ranjan is on the editorial boards of top quality international journals including the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE CLOUD COMPUTING, and *Future Generation Computer Systems*.