International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.141, September 2026

Robust Data Authentication Code for IoT Device Data Authentication in a Layered E-Healthcare System

Punam Prabha, PhD
Department of Computer Science & Engineering
Bhagalpur College of Engineering, Bhagalpur
813210, Bihar, India

Kakali Chatterjee, PhD
Department of Computer Science & Engineering
National Institute of Technology Patna
800005, Bihar, India

ABSTRACT

Data authentication faces severe difficulties in the layered architecture of the e-healthcare system (EHS), with risks to integrity, security, and authenticity occurring at every layer. Conventional authentication techniques often fall short in ensuring comprehensive mes- sage integrity and authenticity. In this context, introducing a robust data authentication code (RDAC) offers a strategic solution for securing healthcare information as it traverses IoT devices. RDAC works in two phases: encapsulation and decapsulation. In the en- capsulation phase, encapsulation authentication code mac1 is generated which is again validated to decapsulation authentication code mac2 in the decapsulation phase. The implementation of RDAC at the internetwork layer promises significant enhancements in security and integrity, ensuring trustworthiness and immutability of data across IoT devices. RDAC exhibits significant gains in performance parameters when compared to current solutions, such as low compilation time, small output size, economical memory usage, efficient resource use, and remarkable operational speed.

Keywords

Decapsulation, e-healthcare system (EHS), encapsulation, privacy, robust data au- thentication code (RDAC), security

1. INTRODUCTION

Advancements in information technologies have transformed the conventional healthcare system into smart healthcare system, integrating the Internet of Thing (IoT), cloud computing, and artificial intelligence [1],[2],[3],[4]. E-healthcare system (EHS) have gained popularity due to their cost-effectiveness, rapid responsiveness, and efficient data storage and sharing mechanisms [5]. Utilizing IoT devices, sensitive healthcare information of users is securely stored in the cloud server, facilitating seamless sharing among healthcare professionals and users [3],[4],[6],[7],[8],[9],[10]. The telecare e-medical service model offer treatment advice based on healthcare monitoring systems [11]. The IoT-based monitoring systems capture patients' physiological parameters through implanted body sensors, transferring the data to the cloud server via an IoT gateway for subsequent sharing with the smart community [2],[3]. IoT devices generate encapsulation and decapsulation key using cryptography to maintain security against different attacks [12],[13],[14].

Analysis and hardware implementation of simplest cryptography algorithm is RC4 stream cipher that emphasize keystream biases [15],[16]. Lightweight stream ciphers for IoT devices compared in [17], focusing on design, performance, and security. However, the analysis is limited to specific lightweight stream ciphers and their suitability for constrained IoT devices. In [18],[19], RC4 vulnerabilities are examined through tracking attacks and keystream distinguishability. It highlighted full-size RC4 which remains secure against practical attacks but requires further improvement for reduced keyspaces. In [20], RC4+ vulnerabilities are demonstrated through distinguishing and differential fault attacks, which may not cover all malicious activities . In [21], a hardware accelerator for RC4+ is presented and proposes pipeline structure. Moreover, it focuses on the trade-off between security and performance. In [22], RC4+ vulnerabilities is analyzed using distinguishing attacks. It suggest that security can be improved by modifying the pad parameter. In [23], Knuth-Morris-Pratt (KMP) algorithm is modeled, focusing on its simplicity and adaptability across pattern matching algorithms. KMP algorithm is implemented and analyzed for string searching, highlighting its linear running time $O(n)$, but limited by the condition $m \leq n$ [24].

The risk of compromising data confidentiality arises during transmission. It creates challenges of integrity, authenticity, immutability, security, and trustworthiness to prevent any compromise of sensitive information [7]. In addition, the complexity of data poses challenges for seamless data sharing and communication, leading to additional issues such as speed during communication, memory consumption, and resource utilization [25]. Robust authentication measures are essential for accessing sensitive patient data [26]. There are various authentication frameworks available in the literature [27],[28],[29],[30]. Most of these authentication frameworks encounter a high false rejection rate, resulting in delays during data authentication. In this context, IoT with a message authentication code has the capability to address the situation. However, the existing solutions [1],[3],[7],[20],[29] are not adaptable to handle both cases while simultaneously maintaining integrity, authenticity, immutability, security, and trustworthiness.

Addressing these issues involves the adoption of blockchain technology, which effectively reduces network communication latency and provides rapid response times [3],[31],[32]. In this context, a layered architecture for EHS is designed to enhance security and privacy of patient data [33]. It comprises four layers namely cloud

layer, blockchain layer, internetwork layer, and system layer. To address robust authentication requirements, the present work introduces robust data authentication code (RDAC) within the sublayers of internetwork layer. It allow the processing of healthcare data in two distinct phases. Initially, data is collected from the blockchain layer and undergoes an encapsulation process to generate framed data. Subsequently, in the second phase, extensive data collected from the communication channel is subjected to a decapsulation process and forwarded to the system layer. The proposed RDAC effectively tackle challenges by efficiently managing large output, enabling faster computing, and ensuring the nominal time required with full utilization of resources.

1.1 Motivation

With the advent of the COVID-19 outbreak, hospitals have increasingly adopted Internet of Medical Thing (IoMT) based remote monitoring frameworks. The purpose of IoMT is to transform traditional medical services into telemedicine, ensuring the safe execution of their responsibilities. The patient data captured and transferred to the health IoT devices for treatment purposes is under the vigilant monitoring of medical practitioners. Furthermore, these data are accessible to entities such as research organizations, pharmaceutical companies, and insurance companies. Securing access to patient data is paramount to prevent vulnerabilities to cyber attacks, which may result in eavesdropping, data breaches, and Denial of Service. Cyber criminals are motivated to secure data, including patient medical records, family health records, bank records, and insurance records. The leakage of such sensitive data poses a substantial threat to patient data safety, privacy, and the confidentiality of information within medical institutions.
In mitigating these risks, it is imperative for IoT sensor nodes to undergo multiple sub-layers network, followed by authentication of their validity to the EHS before transferring patient file. The challenges arise from resource constraints and the vulnerability of physically capturing sensor nodes deployed in EHS environments, making the implementation of a robust authentication protocol particularly challenging, especially in IoT networks. These challenges serve as the driving force behind the motivation to develop a robust authentication scheme for IoT-based EHS.

1.2 Objective

In the present research endeavor, the primary focus is twofold: first, the integration of a data authentication code to validate the authorization of IoT devices; and second, the deployment of a secure hashing algorithm to safeguard the privacy of sensitive patient files (pf) when shared among IoT nodes within an EHS.
To achieve these objectives, a comprehensive framework is designed, encompassing a secure EHS featuring the integration of encapsulation, decapsulation, simple stream cipher and pattern matching algorithm that creates robust data authentication scheme. This framework is strategically engineered to address challenges related to data authentication and security of patient file.

1.3 Contributions

The noteworthy contributions of this research work are pointed out below as:

—To organize IoT devices into distinct sub-layers in internetwork layer and employing robust data authentication code (RDAC) in EHS.

—To employ hashes in RDAC to mitigate the risk of unauthorized access, thereby enhancing the overall robustness of the security measures in EHS. Additionally, a new metric is proposed for robustness using (1).

—To ensure the effectiveness and resilience of the proposed scheme, a security analysis is systematically introduced, meticulously addressing and satisfying all the design measures.

—Furthermore, a comprehensive performance comparison between the proposed scheme and existing alternatives is conducted, evaluating their security features as well as assessing speed, execution time, size of output, memory requirements, and utilization of resources for a thorough understanding of their relative efficiencies.

The rest of the paper is organized as follows: Section 2 discusses the background for the development of the proposed scheme. Section 3 discusses the proposed RDAC scheme employed in the layered architecture of internetwork layer. Moreover, in- detail discussion on security analysis is done in Section 4. Section 5 illustrates the implementation results of the proposed scheme. Finally, the conclusive remarks are given in Section 6.

2. BACKGROUND

This section encapsulates the layered architecture of EHS, exploring potential vulnerabilities and discussing security threats, along with an examination of related work in the relevant field.

2.1 RDAC based E-healthcare system (EHS)

EHS plays a pivotal role in medical imaging, clinical administration, and medical intelligence devices. The general architecture of EHS is structured into four layers: system layer, internetwork layer, blockchain layer, and cloud layer, respectively, as sequenced in lower to upper order (as depicted in Figure 1). The lowest layer is predominantly occupied by the healthcare community, including doctors, hospitals, and lab assistants. The second, internetwork layer, employs network devices such as healthcare and wellness apps, IoT enabled medical equipment, connected medical implants, and wearable fitness trackers for encapsulation and decapsulation. It serves as an intermediary between the system layer and the third layer namely, blockchain layer. The uppermost, cloud layer establishes by the cloud service provider. It acts as the interface where data is securely stored. The articles [33], [34], and [35], respectively, addressed the security issue of medical records in the system layer, blockchain layer, and cloud layer. Thus, the current study addresses the internetwork layer security issue.
In internetwork layer, the data processing begins with the capture and collection of data from patients using physical devices, forming a small data collection network to gather health-related fitness statistics. Subsequently, the captured data is converted into digital format and securely stored in the cloud, making it accessible to stakeholders such as doctors, nurses, emergency services, and clinics based on their specific needs. As shown in Figure 1, the internetwork layer is organized into four sub-layers: (i) Application layer (AL), (ii) Data transport layer (DTL), (iii) Communication network layer (CNL), and (iv) Hardware interface layer (HIL). IoT devices exist in all sub-layers of the internetwork layer. The information fetched at IoT devices is collected by the corresponding sub-layers of the internetwork layer and appends header during the encapsulation phase. Each header is removed by corresponding sub-layers during decapsulation phases. These two phases are used to provide RDAC to verify the authenticity of IoT devices situated at the internetwork layer for transferring patient file (pf) information from the blockchain layer to the system layer.

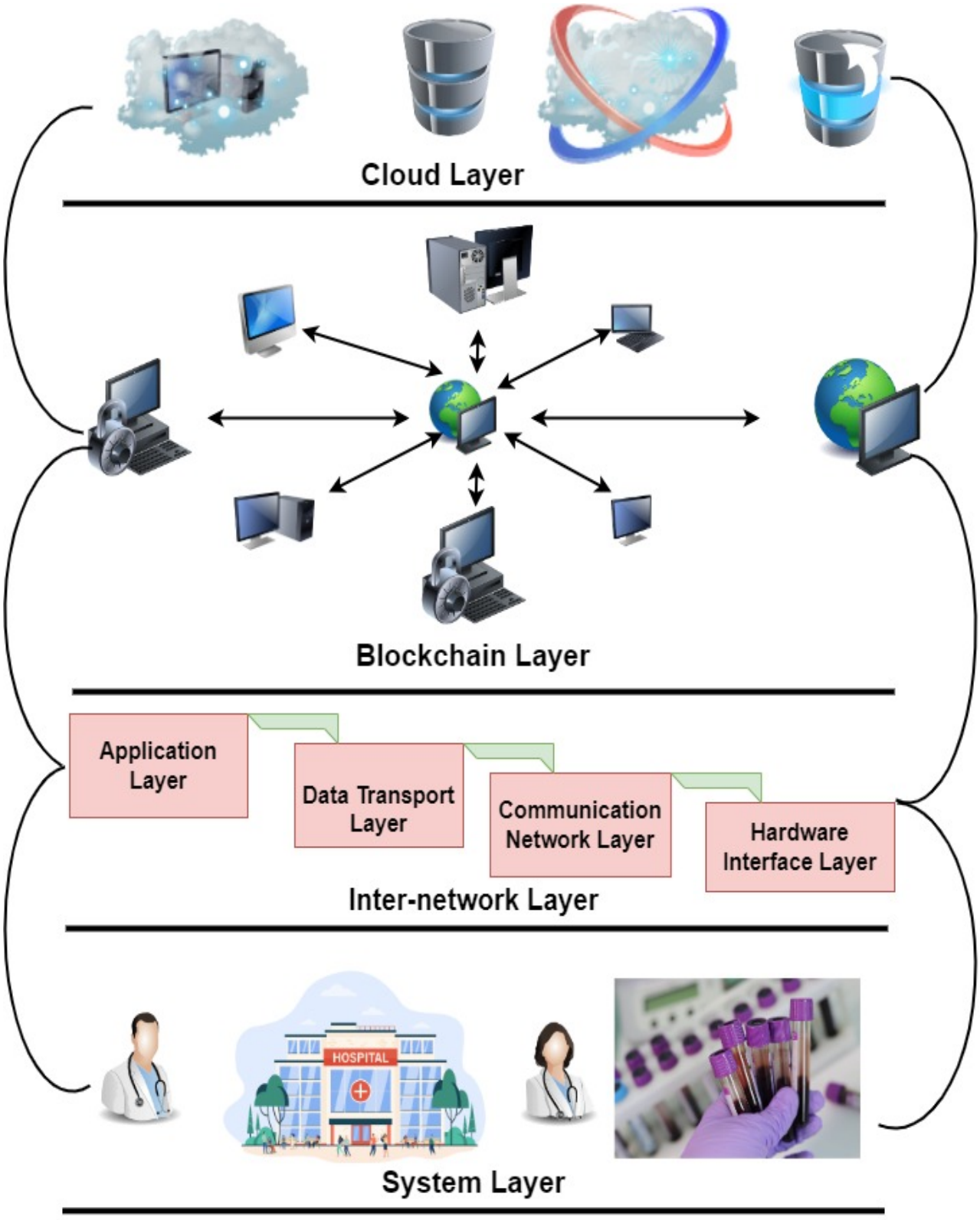


Fig. 1. Layered architecture of e-healthcare system.

2.2 Related work

Numerous studies have been conducted in various domains to identify the aims, tools used, and limitations behind the proposals of the present work, as shown in Table 1. Wang et al. implemented a blockchain-based data storage model for securely storing healthcare data on cloud storage in encrypted form. The metadata of the data is then recorded on a blockchain to ensure and maintain the integrity of the stored information [6].

EHS is featured in various smart projects, with a primary focus on aspects such as IoT integration, forecasting efficiency, network implementation design, security and privacy considerations [27]. The author has concentrated on addressing security and privacy concerns within the fog computing environment by introducing authentication scheme and key management scheme, utilizing a cryptographic hash function and bitwise exclusive-OR operation. Further contributions by Wazid et al. in [30], a lightweight authentication mechanism employing cryptographic hash functions and bitwise exclusive-OR enhanced the security of information collected and transmitted by IoT sensors in surveillance, guarding against potential leaks. The paper published by Shreya et al. in [2] includes a lightweight authentication technique with a privacy-preserving schema utilizing fully homomorphic encryption. This innovative approach not only ensures secure access to patient data online but also allows encrypted data sharing for the intended purposes of stakeholders . In [28], Sun et al. introduced a device-to-server authentication mechanism within the Internet of Medical Thing (IoMT) environment for security and privacy. They presented a hybrid scheme incorporating medium access control and enhanced on-demand vector-enabled routing schemes. While the authentication mechanism was based on two-factor authentication, there is room for improvement.

The paper by Hooshmand et al. [12], deployment in diverse environments introduces a key encapsulation mechanism based on polar codes with secure and efficient key generation, encapsulation, and decapsulation algorithms that reduce the ephemeral secret decapsulation key size and maintain security against various attacks, but it has limited analysis on scalability. The paper by Yang et al. [1] surveys IoT security and privacy challenges, emphasizing rapid technological changes, practical implementation constraints, incomplete data, and evolving threats. This survey consists of four segments including most relevant limitations of IoT devices and their solutions, classification of IoT attacks, mechanisms and architectures for authentication and access control, and analyze the security issues in different layers. Similarly, Jurcut et al. [8] survey IoT security, addressing broad issues but facing limitations like rapid changes, data gaps, practical constraints, and potential bias. Khan et al. [3] look into IoT security problems in general but lacks in detail analysis. Son et al. [10] tried to improve the security of IoT identification, but scalability could be a problem.

The aim of Badhib et al.'s paper [26] is to develop a fast and secure device-to-device continuous authentication protocol for the IoT, addressing impersonation and DoS attacks using devices' features and mitigating threats with shadow IDs and emergency keys, but the limitation lies in scalability, diverse IoT environments, real-world deployment, and computational constraints. The aim of Farzam et al.'s paper [14] is to develop a high-speed supersingular isogeny Diffie-Hellman and key encapsulation, focusing on optimizing large-degree isogeny computations and field arithmetic operations to significantly enhance performance.

In [5], Wei et al.'s developed an efficient, secure, and privacy-preserving message authentication scheme for IoT, addressing data security and privacy concerns, while the limitation may involve challenges in scalability and real-world deployment due to diverse IoT device constraints and computational requirements. In [36], created a secure hyperledger fabric-based blockchain architecture for managing electronic health records while ensuring privacy. The necessity for specialized blockchain infrastructure and expertise leads to high deployment and maintenance complexity. In Lee et al.'s paper [13], GPU-accelerated techniques was proposed to enhance Kyber, a post-quantum Key Encapsulation Mechanism, enabling high throughput performance crucial for securing IoT sensor data. However, frequent key generation and secure transmission to gateway devices and cloud servers remain time-consuming challenges. In [37], Gayathri et al. created a blockchain-based secure EHS that uses incremental Provable Data Possession, Zero Exponential Mikhail's Hash Knowledge Proof, Transport Layer Security, Homomorphic Encryption, Thompson's Extended Transition Construction Algorithm for secure electronic health record sharing, full node authentication, and data integrity. However, the method has higher computational complexity and authentication latency because it uses several cryptographic mechanisms.

In [38] improved construction safety compliance by utilizing blockchain,IoT sensors, and smart contracts for decentralized and tamper-proof workforce management. However, the approach wss constrained by the oracle problem, consensus protocol scalability issues, and the lack of legal recognition and interoperability with BIM standards. The aim of Bera et al.'s paper [16] is to investigate and improve the cardiology department's operational system flexibility using simulation models, though this requires extensive data, high computational resources, and complex model validation.

The above discussion has identified a research gap in the existing literature.

—Despite the myriad advantages of EHS, a notable challenge persists in the form of insufficient integrity, security, immutability, authenticity and trustworthiness for patient data transmit in the public cloud.

—The absence of a robust authentication mechanism poses a challenge in effectively authenticating IoT devices responsible for handling patient file.

—Frequently, authentication-based schemes encounter challenges such as complex cryptographic technique, high speed, less utilization of resources, and huge storage, leading to potential shortcomings.

3. PROPOSED FRAMEWORK

The proposed robust data authentication code (RDAC) is used to authenticate Internet of Things (IoT) devices by encapsulating and decapsulating data using a straightforward and effective cryptographic technique that requires less speed, full utilization of IoT devices, and less storage. This section provides a detailed description of the implementation of RDAC in the internetwork layer of layered EHS.

The implementation of the proposed RDAC based data authentication technique makes use of certain assumptions discussed below as:

—All stakeholders of EHS must register at the system layer, undergo verification at the blockchain layer via the internetwork layer, and finally be deployed under the cloud layer.

—RDAC is implemented mainly in three layers: the blockchain layer, the internetwork layer, and the system layer of EHS.

—The patient file information (pf) is accessible to the hardware interface layer (HIL).

Table 1. A literature survey on authentication schemes in EHS.

Paper	Year	Domain	Aim	Tool	limitation
[1]	2017	IoT	Systematic examination of the security and privacy issues	Survey consists of four segments	Incomplete data, evolving threats.
[3]	2018	IoT	Survey on IoT security issues.	Threat categorization, protocol analysis, blockchain discussion, solution mapping.	Lead to superficial treatment of specific IoT security issues.
[39]	2018	IoMT	Detect real or fake medical images	Enhancing security beyond unimodal biometric system	Security analysis missing
[29]	2019	IoT	Lightweight construction and lower network latency	Identity management and user authentication on a permissioned blockchain	Comparision analysis missing
[40]	2019	IoMT	Security and privacy of IoMT	IoMT using Secure Assessment Framework	Complex mechanism
[8]	2020	IoT	Survey on IoT security threats	Security threats, identify weaknesses, present solutions, mitigate risks	Broad scope, rapid changes, data gaps, potential bias.
[5]	2020	IoT	Efficient, secure, and privacy-preserving message authentication scheme.	Cryptographic offline or online computation	Challenges in scalability and deployment of IoT security scheme.
[14]	2020	EHS	Implementation of supersingular isogeny Diffie-Hellman and key encapsulation	Montgomery Ladder Technique.	High computational cost.
[9]	2020	EHS, IoT	Innovative three-factor lightweight authentication scheme	Symmetric encryption	Result analysis missing
[41]	2020	IoMT	Security of information exchanged among sensor nodes	Mutual authentication and secret key establishment	Lacks value in adverse situation
[26]	2021	IoT	Fast and secure device-to-device authentication.	Informal and formal test analysis.	Real-world deployment, and computational constraints.
[10]	2021	IoT	Secure and efficient IoT authentication scheme addressing existing vulnerabilities.	Informal analysis and formal analysis.	Scalability issues and real-world deployment challenges.
[13]	2021	IoT	GPU-accelerated Kyber enhancements for high throughput IoT sensor data security.	Rejection sampling, central binomial distribution, and fine-grain AES-256.	High computational and knowledge requirements.
[31]	2021	EHS	Adaptation of Attribute-based Searchable Encryption (ABSE)	Addressing the significant computational costs of the ABSE scheme	Data about correctness, robustness missing
[12]	2022	EHS	Secure, efficient key generation, encapsulation, decapsulation for efficient cryptographic operations.	Polar coding.	Limited analysis on scalability and practical deployment.
[2]	2022	IoMT	Lightweight authentication privacy-preserving scheme	Homomorphic encryption	Complicated mechanism
[16]	2023	EHS	Cardiology department's operational system to improve flexibility EHS	Simulation models.	complex model validation.
[36]	2024	IoT	Hyperledger Fabric blockchain architecture for EHR management.	Hyperledger Fabric.	High deployment and maintenance complexity.
[38]	2025	EHS	To enhance construction safety compliance	decentralized and tamper-proof workforce management	limited by the oracle problem.
[37]	2026	EHS	Develop a blockchain-based secure Electronic Healthcare System	Homomorphic encryption	High computational overhead and increased authentication latency.

—The internetwork layer allows for the smooth flow of verified patient file (pf) information between the blockchain layer and the system layer within EHS.

—IoT devices are appropriately placed in sub-layers of the internetwork layer to collect medical data.

The symbols and abbreviations employed in the present work are enlisted in Table 2.

3.1 Internetwork layer architecture

The layered architecture of the internetwork layer is shown in Figure 2. Each sub-layer contains IoT devices with distinct roles and capabilities. Figure 3 illustrates the operational functionalities of the proposed RDAC in EHS, described in detail below as:

—The first stage involves the internetwork layer retrieving verified healthcare stakeholder information from the blockchain layer.

—The application layer (AL), the topmost sub-layer of the internetwork layer, is used to program healthcare and wellness apps (hwa) using this information.

—The data transport layer (DTL) receives the information hosted on the health and wellness apps (hwa).

—The data transport layer (DTL) appends additional information that is available on IoT enabled medical equipment (eme).

—The data is gathered by the communication network layer (CNL) and processed by RDAC to provide the hardware interface layer (HIL) relevant information about connected medical implants (cmi).

—The hardware interface layer (HIL) generates encapsulation authentication code ($mac1$) once it has collected all the necessary attribute data, including that from connected medical implants (cmi), patient file information (pf), and wearable fitness tracker (wft).

—Once the encapsulation procedure is complete, the communication channel is loaded with encapsulation authentication code ($mac1$) and patient file information (pf).

—In the initial stages of decapsulation, the communication channel is utilized by the wearable fitness tracker (wft) to gather encapsulation authentication code ($mac1$) and patient file information (pf).

—Any wearable fitness tracker (wft) can be accessed using the hardware interface layer (HIL).

—The communication network layer (CNL) generates connected medical implants (cmi).

Table 2. Nomenclature of codes.

Code	Acronyms
hwa	health and wellness app
eme	IoT enabled medical equipment
cmi	connected medical implants
wft	wearable fitness tracker
pf	patient file
AL	application layer
DTL	data transport layer
CNL	communication network layer
HIL	hardware interface layer
IL	inter-network layer
$mac1$	encapsulation authentication code
$mac2$	decapsulation authentication code
$\ll$	left shift
$\gg$	right shift
$\oplus$	XOR operation
$+$	binary addition
&	and
[]	array index
%	reminder operator
$\|\|$	concatenation
$\leq$	less than equal
$\neq$	not equal

—The data transport layer (DTL) computes decapsulation authentication code ($mac2$) with IoT enabled medical equipment (eme).

—Comparing encapsulation authentication code ($mac1$) to decapsulation authentication code ($mac2$) the application layer (AL) stores final result on health and wellness apps (hwa).

—If all of the above is correct, the message is considered authentic, and the patient file (pf) is sent to the system layer; if it isn't, it means there's a problem with their originality.

3.2 Robust data authentication code (RDAC)

In the encapsulation phase, IoT device generate hash1 h1() and call Algorithm 1 to get encapsulation authentication code $mac1$ using Algorithm 2. During the decapsulation phase, IoT devices implements Fluhrer Mantin and Shamir (FMS) cryptanalysis technique [42] using Algorithm 3, and compute decapsulation authentication code $mac2$. As a result, Algorithm 4 is used to match encapsulation authentication code $mac1$ and decapsulation authentication code $mac2$ using Knuth Moriss Pratt (KMP) pattern matching algorithm [43]. Following the completion of the encapsulation and decapsulation processes, RDAC authenticates medical information among internetwork nodes using Algorithm 5.

3.2.1 Data encapsulation process. The algorithm 1 generates hash1 h1(), which is used in Algorithm 2 (line 3). Algorithm 2 describes the entire message flow for the first phase (data encapsulation) of the authentication procedure.

—In the first step, the internetwork layer receives verified patient file (pf) information from the blockchain layer.

—The final added information is assigned to the application layer (AL), which contains the data available through health and wellness apps (hwa) and the verified patient file (pf).

Algorithm 1: Hash1 h1() used in Encapsulation

Input: AL
Output: DTL
Data: pf,hwa
1 $i = 0$
2 j=0
3 AL=pf +hwa
4 l=$AL.length$
5 let s[] be new array
/* KSA (Key Scheduling Algorithm) */
6 **for** i=*0 to* $n-1$ **do**
7 $s[i]$=i
8 **for** i=*0 to* $n-1$ **do**
9 j=j +$s[i]$ +$AL[i]$
10 $s[i]$=$s[i]$ + $AL[i]$
11 $AL[i]$=$s[i]$- $AL[i]$
12 $s[i]$ =$s[i]$ - $AL[i]$
/* RC4PRGA1 */
13 **for** i=*0 to* $n-1$ **do**
14 i=$(i+1)$ % n
15 j=(j+s[i]) % n
16 **for** i=*0 to* $n-1$ **do**
17 $s[i]$=$s[i]$ + $AL[i]$
18 $AL[i]$=$s[i]$- $AL[i]$
19 $s[i]$ =$s[i]$ - $AL[i]$
20 $DTL1$= (s[$s[i]$ + $s[j]$]) % n
/* RC4++PRGA2 */
21 **for** i=*0 to* $n-1$ **do**
22 i=$(i+w)$ % n
23 a=$s[i]$
24 j=$(j+a+w)$ % n
25 b=$s[j]$
26 $s[i]$=b
27 $s[j]$=a
28 **for** i=*0 to* $n-1$ **do**
29 $s[i]$=$s[i]$ + $AL[i]$
30 $AL[i]$=$s[i]$ - $AL[i]$
31 $s[i]$ =$s[i]$ - $AL[i]$
32 c = (($s[i \ll 5 + j \gg 3)\%2] + (s[j \ll 5 + i \gg 3)\%2]) + 0XAA)\%2$
33 $DTL2$= (s[$s[a+b]$ + $s[c]$] + $s[j+b]$)%2
34 w=w + 2
/* RC4++PRGA3 */
35 **for** i=*0 to* $n-1$ **do**
36 i=$(i+w)$ % n
37 a=$s[i]$)
38 j=$(j+a+w)$ % n
39 **for** i=*0 to* $n-1$ **do**
40 $s[i]$=$s[i]$ + $AL[i]$
41 $AL[i]$=$s[i]$- $AL[i]$
42 $s[i]$ =$s[i]$ - $AL[i]$
43 c = (((s[i \ll 5 + j \gg 3)\%2] + (s[j \ll 5 + i \gg 3)\%2]) + 0XAA)\%2$
44 $DTL3$= (s[$s[a+b]$ + $s[c]$] + $s[j+b]$)%2
45 w=w + 2
/* RDAC_PRGA */
46 $DTL = DTL1 + DTL2 + DTL3$

Algorithm 2: Phase 1: Data Encapsulation

Input: pf
Output: $mac1$
Data: hwa, eme, cmi, wft
1 IL =pf
2 $AL = pf \,\|\, hwa$
3 eme=$h1(AL)$
4 DTL=eme
5 CNL=$cmi \oplus DTL$
6 HIL=$wft \,\|\, CNL$
7 $mac1$= CNL
8 **return** HIL

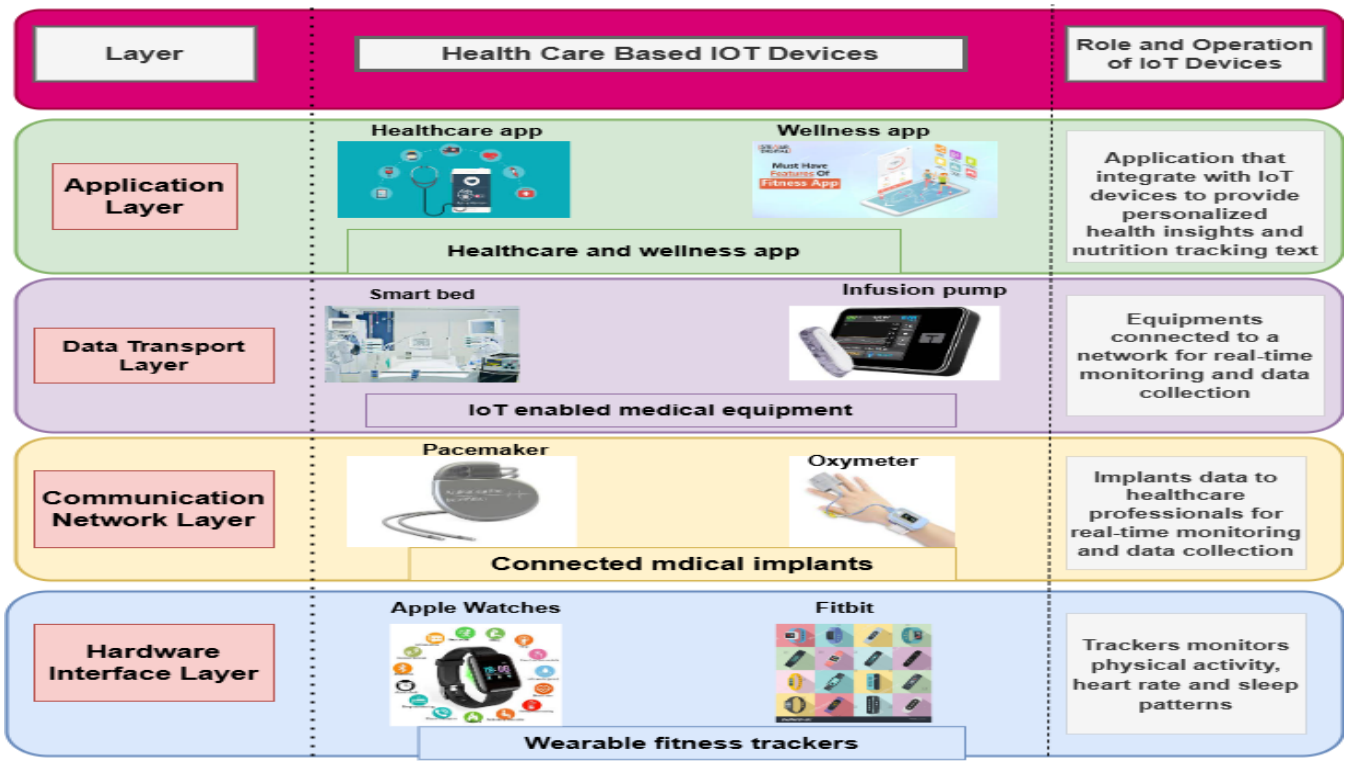


Fig. 2. Layered architecture of internetwork layer.

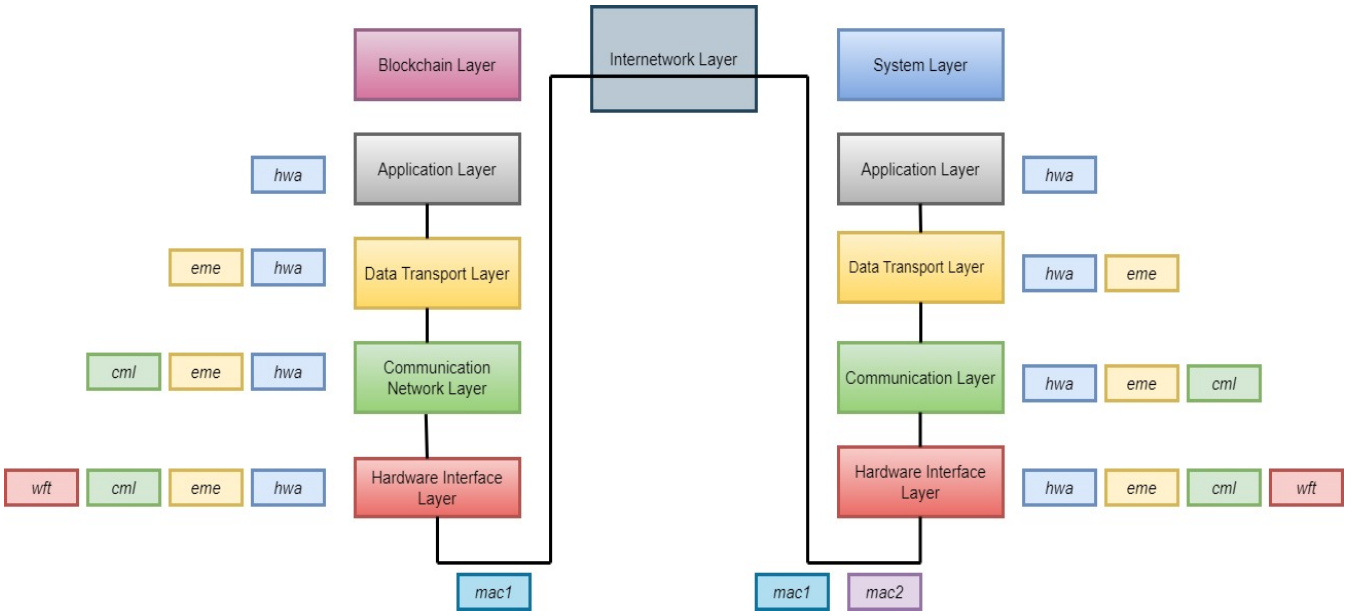


Fig. 3. Encapsulation and decapsulation phases of RDAC based authentication scheme.

—When IoT enabled medical equipment (eme) processes the application layer (AL), then Algorithm 1 is used for hashing.

—In this algorithm, RDAC_PRGA is constructed utilizing the finalize result of $DTL1$, $DTL2$, and $DTL3$, which store the RC4 cipher [15], RC4++ cipher [44], and RC4+* cipher [44] respectively.

—The information processed by IoT enabled medical equipment (eme) is now available at the data transport layer (DTL).

—The communication network layer (CNL) combines all information from the data transport layer (DTL) and performs an XOR operation on data collected by connected medical implants (cmi).

—The final added information, which includes data from both the communication network layer (CNL) and the wearable fitness tracker (wft), is assigned to the hardware interface layer (HIL).

—Encapsulation authentication code ($mac1$) is used to hold data from the communication network layer (CNL).

—The communication link is where the content of the hardware interface layer (HIL) is located.

3.2.2 Data decapsulation process. During this phase, Hash2 h2() is computed based on cryptanalysis employed in [42] as seen in Algorithm 3. The calculated result is compared to the achieved result using KMP [43] as shown in Algorithm 4. Algorithm 5 provides a thorough overview of message flow during the authentication process's decapsulation phase.

Algorithm 3: Hash2 h2() used in Decapsulation

Input: HIL
Output: cmi
Data: wft
1 HIL=wft
2 j=0
3 l=$HIL.length$
4 let s[] be new array
5 **if** $s[i] \leq l$ **then**
6 goto step 8
7 **else if** $(s[i] + s[s[i]]) \equiv 1$ **then**
8 goto step 10
9 **else if** $s^{-1}cmi[i-1] \neq 1$ **then**
10 goto step 12
11 **else if** $s^{-1}cmi[i-1] \neq s[1]$ **then**
12 goto step 16
13 **else**
14 goto step 21
15 **for** i=*0 to* l **do**
16 j=(j+s[i]) % l
17 cmi[l]= $s^{-1}cmi[i]$ - j - $s[l]$
18 Repeat step 5 to 15
19 **return** cmi

—The initial step assigns the content of the communication link to the wearable fitness tracker (wft).

—Consequently, the wearable fitness tracker (wft) is assigned to the hardware interface layer (HIL).

—The communication network layer (CNL) stores data gathered by connected medical implants (cmi) using the XOR operation on data produced by the Algorithm 4 on the hardware interface layer (HIL).

—Data from the communication network layer (CNL) is available to IoT enabled medical equipment (eme).

—As a result, the data transport layer (DTL) accesses data from IoT enabled medical equipment (eme).

—The data transport layer (DTL) retrieves information and securely stores it as a decapsulation authentication code ($mac2$).

—The health and wellness apps (hwa) are hashed with Algorithm 4, as stated in line 7 of Algorithm 5.

—The application layer (AL) contains the results of health and wellness apps (hwa).

Algorithm 4: Hash3 h3() used in Decapsulation

Input: $mac1$, $mac2$
Output: q
Data: x, y, z
1 **Function** CPF($mac2$):
2 y=$mac2.length$
3 let z[] be new array
4 k=0
5 **while** $k > 0$ & *mac2[k+1]* $\neq mac1$ *[q]* **do**
6 k=$z[k]$
7 **if** $mac2[k+1]$ = $mac1[k]$ **then**
8 k=$k+1$
9 $z[q]$=k
10 **for** q=*2 to* y **do**
11 step 4 to 8
12
13 **Def** KMP($mac1$, $mac2$):
14 x=$mac1.length$
15 y=$mac2.length$
16 z=CPF($mac2$)
17 q=0
18 **while** $q > 0$ & $mac2[q+1] \neq mac1$ *[i]* **do**
19 q=$z[q]$
20 **if** $mac2[q+1]$ = $mac1$ *[i]* **then**
21 q=$q+1$
22 **if** q= y **then**
23 Pattern occur with shift $q - y$
24 q=$z[q]$
25 **for** i=*1 to* x **do**
26 step 16 to 23
27

—Successful authentication is obtained at the application level (AL) if the encapsulation authentication code ($mac1$) and decapsulation authentication code ($mac2$) show a similar pattern.

Algorithm 5: Phase 2: Data Decapsulation

Input: pf, $mac1$
Output: AL
Data: hwa, eme, cmi, wft
1 wft=$pf \parallel CNL$
2 HIL=wft
3 CNL=$cmi \oplus h2(HIL)$
4 eme=CNL
5 DTL=eme
6 $mac2$=DTL
7 hwa=$h3(mac1, mac2)$
8 AL=hwa
9 **return** AL

4. SECURITY ANALYSIS

In this section, the security features of RDAC is discussed in detail. RDAC guarantees authenticity, integrity, immutability, security, and trustworthiness by implementing an integration of encapsulation, decapsulation, and hash operations. Combining these with blockchain security systems will improve the trustworthiness and robustness of the system even further.

LEMMA 1. *The proposed concept of recognizing integrity makes sure that the IoT devices at the communication network layer (CNL) are not changed while they are being sent or stored.*

PROOF. Algorithms 1, 2, and 5 alter data using $s[]$ array manipulation and numerous rounds of PRGA, ensuring that any unauthorized modification results in inaccurate outputs (see Algorithm 1, lines 20, 33, 44, and 46). The encapsulation phase (see Algorithm 2, line 3) employs hashing h1() to generate a unique IoT enabled medical equipment (eme), which functions as a fingerprint for the input data, allowing for detection of any changes.

Hash function for integrity

$$eme = h1(AL)$$

Verification

$$\begin{aligned} DTL1 &= (s[s[i] + s[j]])\,\%n \\ DTL2 &= (s[s[a+b] + s[c]] + s[j+b])\,\%2 \\ DTL3 &= (s[s[a+b] + s[c]] + s[j+b])\,\%2 \\ DTL &= DTL1 + DTL2 + DTL3 \end{aligned}$$

Integrity validation

$$\begin{aligned} CNL &= cmi \oplus DTL \\ CNL_{new} &= cmi \oplus h2(HIL) \\ \text{Check ; } CNL &= CNL_{new} \end{aligned}$$

The consistent validation of the data transport layer (DTL) and the employment of Algorithm 4 for pattern matching reinforces the data integrity. □

LEMMA 2. *The proposed approach conceals authentication while ensuring that the data and IoT devices involved are genuine.*

PROOF. Algorithm 1 is used to generate a sequence based on input keys and data, guaranteeing that only authentic keys produce the desired sequence. The employment of h1(), h2(), and h3() hash functions in the encapsulation and decapsulation procedures of Algorithms 1, 3, and 4 ensure that only authenticated data can generate valid results. The encapsulation phase comprises methods that combine and hash numerous inputs to confirm that the data is from a genuine source.

Key scheduling algorithm (KSA) initialization

$$\begin{aligned} s[i] &= i \quad \textbf{for } (0 \le i < n) \\ j &= (j + s[i] + AL[i])\%n \end{aligned}$$

PRGA for key stream generation

$$\begin{aligned} i &= (i+1)\%n \\ j &= (j + s[i])\%n \end{aligned}$$

Encapsulation

$$\begin{aligned} AL &= pf \parallel hwa \\ eme &= h1(AL) \\ DTL1 &= (s[s[i] + s[j]])\%n \end{aligned}$$

Authenticity validation

$$\begin{aligned} mac1 = CNL &= cmi \oplus DTL \\ HIL &= wft \parallel CNL \end{aligned}$$

Decapsulation

$$\begin{aligned} CNL &= cmi \oplus h2(HIL) \\ hwa &= h3(mac1, mac2) \\ AL &= hwa \end{aligned}$$

The usage of hash functions $h1()$, $h2()$, and $h3()$ ensures the authenticity of data through robust cryptographic techniques. □

LEMMA 3. *The proposed method conceals immutability, ensuring that once data is written, it cannot be altered.*

PROOF. While algorithms can not guarantee immutability, a combination of hashing and key scheduling can be used to accomplish it. Storing hashes of data representations may assist in ensuring that data has not been altered, hence indirectly enabling immutability.

Blockchain based immutability (if integrated)

$$\begin{aligned} \text{Block}_i &= h(\text{Block}_{i-1}, \text{Data}_i) \\ \text{Blockchain} &= \{\text{Block}_1, \text{Block}_2, \ldots, \text{Block}_n\} \end{aligned}$$

Hash and key scheduling contributions

$$h(AL) = h(pf \parallel hwa)$$

Immutable record

$$\text{Immutable record} = h(\text{previous record, new data})$$

When blockchain technologies are combined, they create a tamper-proof record of data transactions, increasing immutability. □

LEMMA 4. *The suggested method assures data security and includes every safety measure to prevent cyber attacks and unauthorized access.*

PROOF. The usage of KSA and PRGA algorithms for key generation and data transformation provides an additional layer of security against unwanted access. The encapsulation and decapsulation steps include checks and validations to ensure that only genuine, authenticated data is processed.

Key scheduling

$$\begin{aligned} s[i] &= i \quad \textbf{for } (0 \le i < n) \\ j &= (j + s[i] + AL[i])\%n \end{aligned}$$

PRGA operations

$$\begin{aligned} i &= (i+1)\%n \\ j &= (j + s[i])\%n \end{aligned}$$

XOR operation for obfuscation

$$\begin{aligned} CNL &= cmi \oplus DTL \\ HIL &= wft \parallel CNL \end{aligned}$$

Encryption and decryption

$$\begin{aligned} DTL &= DTL1 + DTL2 + DTL3 \\ AL &= pf \parallel hwa \end{aligned}$$

Hash function usage

$$eme = h1(AL)$$
$$CNL_{new} = cmi \oplus h2(HIL)$$

The combined application of KSA, PRGA, and XOR operations results in a secure environment for data encapsulation and decapsulation process. □

LEMMA 5. *The proposed scheme ensures data trustworthiness, which relates to the system's robustness and trustworthiness.*

PROOF. The organized method to data encapsulation and decapsulation, which includes numerous verification phases, improves the system's trustworthiness. The use of common cryptographic procedures, such as hashing and key scheduling, improves the system robustness.

Consistent use of cryptographic techniques

$$eme = h1(AL)$$
$$CNL = cmi \oplus DTL$$
$$hwa = h3(mac1, mac2)$$

Multiple verification steps

$$DTL = DTL1 + DTL2 + DTL3$$
$$HIL = wft \parallel CNL$$

Validation of outputs

$$AL = hwa$$
$$mac2 = DTL$$

Pattern match

$$q = \text{KMP}(mac1, mac2)$$

Trustworthiness metrics

$$\text{Robustness} = \frac{\text{No. of successful validations}}{\text{Total no. of checks}} \quad (1)$$

By guaranteeing the successful validation of the majority of data transactions, robustness quantifies the system's trustworthiness. □

5. RESULT ANALYSIS

This section assesses the performance of the proposed RDAC authentication scheme by examining its CPU usage, processing speed, memory consumption, compilation time, and output size. The RDAC system was developed on a 64-bit operating system with an x64 processor, featuring a CPU speed of 1.60 GHz and 8GB of RAM.

First, the proposed RDAC implementation involves sending the verified patient file pf for the packaging process, as shown in Algorithm 2. The packaging process uses RDAC to create encapsulation authentication code $mac1$ (see Algorithm 1). Then, patient file pf and encapsulation authentication code $mac1$ are saved on the communication link for the decapsulation process, as described in Algorithm 5, with the assistance of FMS and KMP (see Algorithms 3 and 4). Using the C++ programming language, the suggested authentication scheme RDAC is run step by step along with the RC4, RC4++, and RC4+* stream ciphers.

The RC4, RC4++, RC4+*, and RDAC stream ciphers is evaluated to observe which ones use the CPU the most at a normal speed, as shown in Figure 4 and Table 3.

A comparative analysis is done based on speed, CPU usage, and memory storage, as shown in Figure 5. Moreover, Table 4 and Figure 6 show a comparison of how well RC4, RC4++, RC4+*, and RDAC provide robustness using (1) during the process of encapsulation and decapsulation.

To figure out robustness, the changeable size of the $S[]$ array and the fixed size of the patient file (pf) are both taken into considerations. The 8-bit value of encapsulation authentication code $mac1$ is mapped to the 8-bit value decapsulation authentication code $mac2$. In this case, 8-bit $mac1$ = "00000000" needs 320 successful validations, with checks for RC4 (95), RC4++ (140), RC4+* (39), and RDAC (13). This means that the system is robust when using RC4 (3.37), RC4++ (2.29), RC4+* (8.21), and RDAC (24.62) because RDAC is superior robustness value in compare to RC4, RC4++, and RC4+* (see Table 4).

6. CONCLUSION AND FUTURE WORK

Ensuring secure authentication in layered architecture of EHS is imperative, especially when verified patient file details is passing through blockchain layer to system layer through internetwork layer. Data authentication in sophisticated EHS is difficult due to integrity, security, and authenticity risks across layers. Traditional authentication methods rarely guarantee communication integrity and authenticity. Implementation of robust data authentication code (RDAC) at the internetwork layer improves security and integrity, making data across networks trustworthy and immutable. RDAC outperforms earlier methods in compilation time (0.24 sec), output size (137.333 KiB), memory usage (47%), resource utilization (80%), and operational speed (3.29 GHz) (see Table 3). As a result, RDAC makes full use of IoT devices at a lower speed, requires very little memory, and uses straightforward cryptographic techniques like RC4, RC4+, and RC4++.

Future research will investigate the performance of the RDAC in large-scale IoT environments with diverse device configurations. It will also assess its effectiveness and reliability in real-world healthcare settings, its ability to defend against emerging and advanced security threats, its compatibility with existing IoT-based EHS platforms, and its potential applications.

Declaration

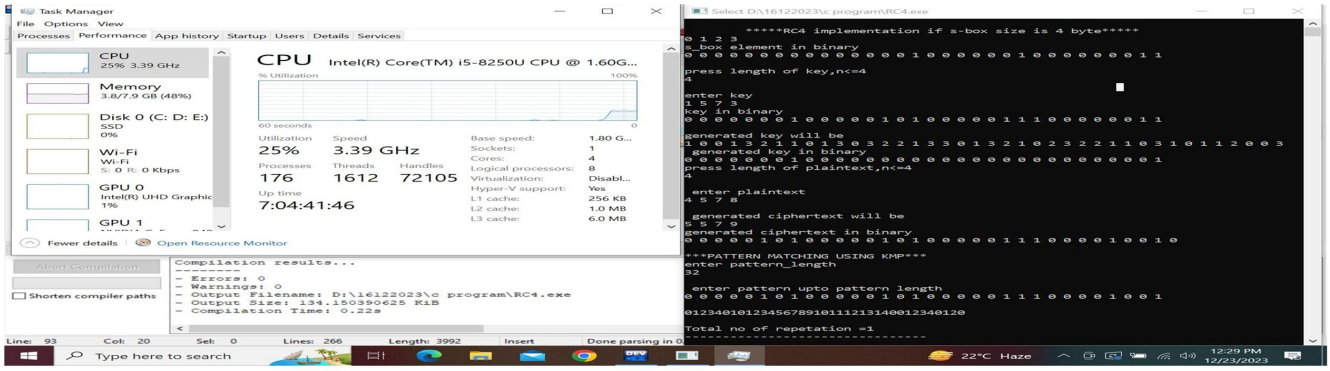
—Ethical approval: Not applicable

—Competing interests: The authors declare that they have no known competing financial interests or personal relationship that could have appeared to influence the work reported in this paper.

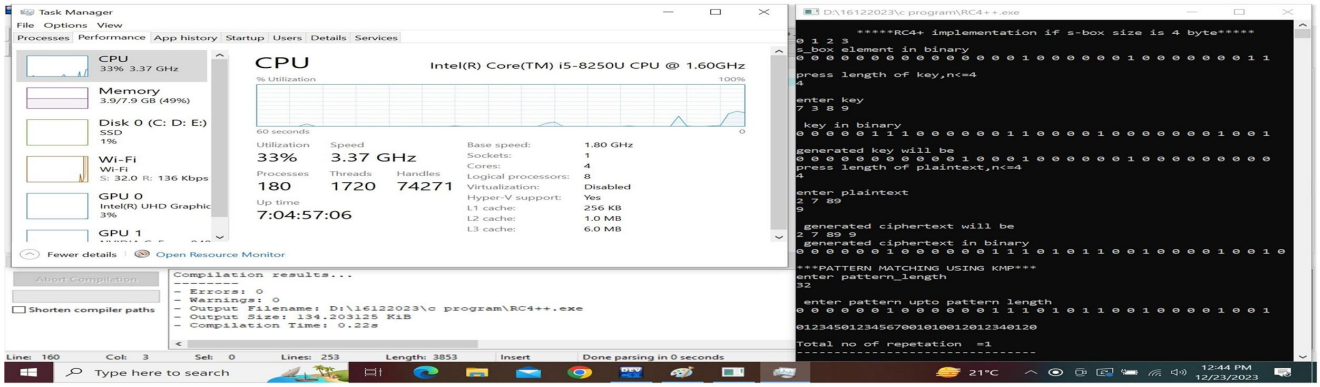
—Authors' contributions: The authors confirm contribution to the paper as follows: Punam Prabha: Data curation, Formal analysis, Writing-original draft, Conceptualization, Writing-review & editing, Validation. Kakali Chatterjee: Writing-review & editing, Validation

—Funding: Not applicable

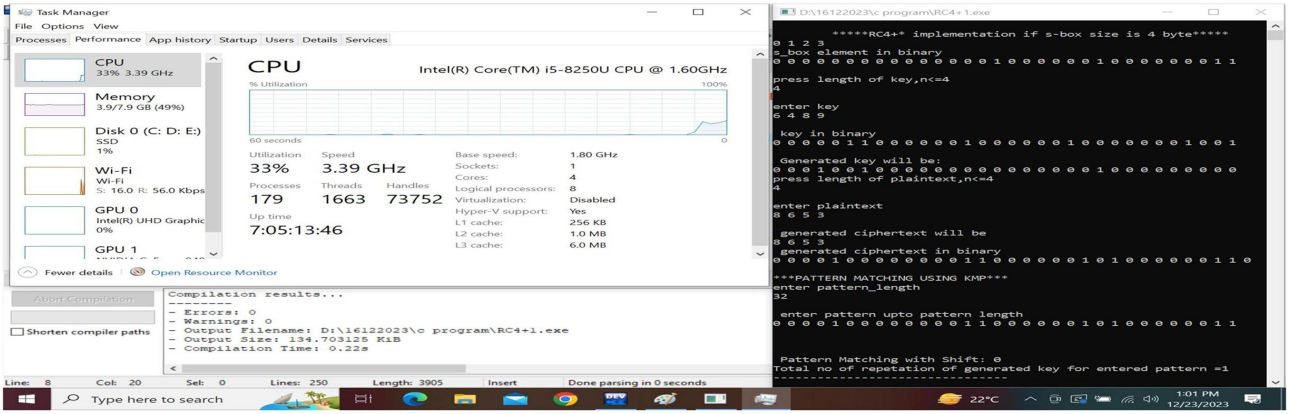
—Availability of data and materials: Not applicable



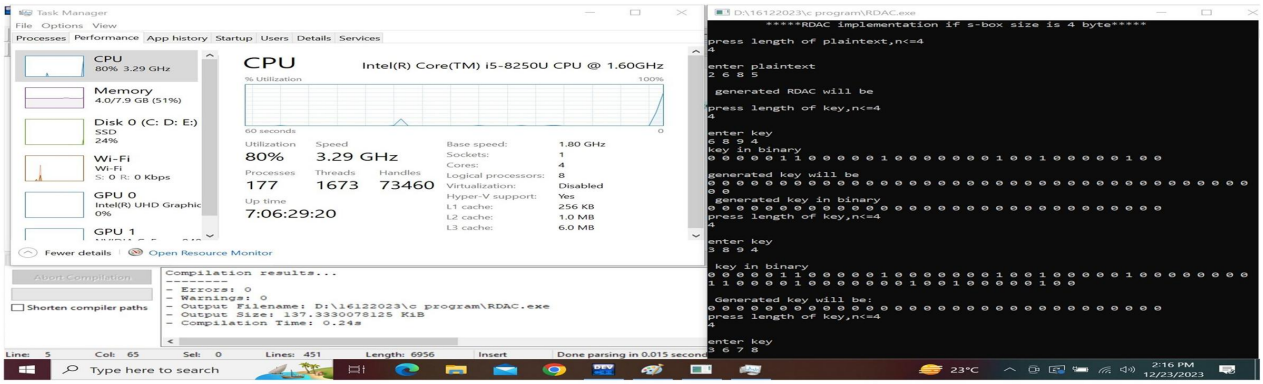
(a) RC4



(b) RC4++



(C) RC4+*



(d) RDAC

Fig. 4. Execution of (a) RC4 (b) RC4++ (c) RC4+* and (d) RDAC.

Table 3. Comparative analysis of the proposed method with existing method.

Paper	Hash1()	CT (sec)	output (KiB)	utilization (%)	speed (GHz)	memory (%)
[19]	RC4	0.22	134.150390625	25	3.39	48
[44]	RC4++	0.22	134.203125	33	3.37	49
[44]	RC4+*	0.22	134.703125	33	3.39	49
Proposed	RDAC	0.24	137.3330078125	80	3.29	51

Table 4. Robustness with various $s[]$ array.

$mac1$ (8 bit)	validation (count)	Check				Robustness			
		RC4	Rc4++	RC4+*	RDAC	RC4	Rc4++	RC4+*	RDAC
s[]=4 byte									
00000000	320	95	140	39	13	3.37	2.29	8.21	24.62
00000001	320	19	13	11	9	16.84	24.62	29.09	35.56
00000000	320	95	140	39	13	3.37	2.29	8.21	24.62
00000001	320	19	13	11	9	16.84	24.62	29.09	35.56
00000010	320	19	12	9	7	16.84	26.67	35.56	45.71
00000011	320	9	2	4	1	35.56	160.00	80.00	320.00
s[]=8 byte									
00000100	320	16	10	10	9	20.00	32.00	32.00	35.56
00000101	320	5	3	2	2	64.00	106.67	160.00	160.00
00000110	320	14	6	4	2	12.86	53.34	80.00	160.00
00000111	320	2	1	2	1	160.00	320.00	160.00	320.00
s[]=16 byte									
00001000	320	9	10	7	4	35.56	32.00	45.71	80.00
00001100	320	5	13	7	1	64.00	24.62	45.71	320
00001010	320	7	2	1	1	45.71	160.00	320.00	320.00
00001011	320	5	1	5	3	64.00	320.00	64.00	106.67

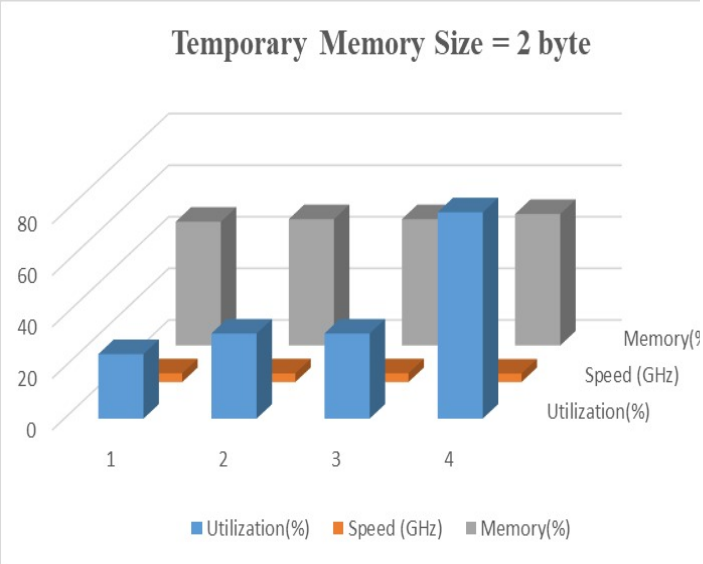


Fig. 5. Performance graph of proposed protocol with existing protocol.

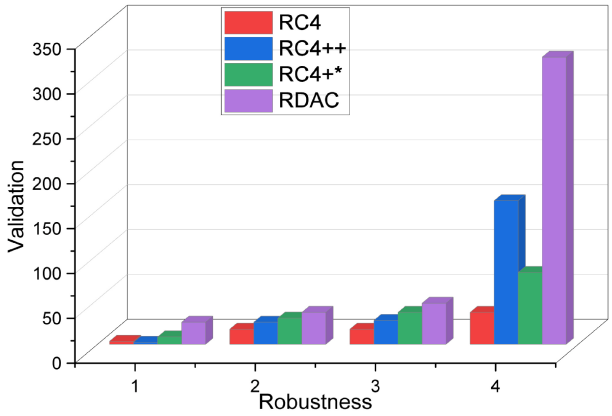


Fig. 6. Robustness of RDAC having s[]=4 byte.

7. REFERENCES

[1] Yuchen Yang, Longfei Wu, Guisheng Yin, Lijie Li, and Hongbin Zhao. A survey on security and privacy issues in Internet-of-Things. *IEEE Internet of things Journal*, 4(5):1250–1258, 2017.

[2] Shashi Shreya, Kakali Chatterjee, and Ashish Singh. A smart secure healthcare monitoring system with Internet of Medical Things. *Computers and Electrical Engineering*, 101:107969, 2022.

[3] Minhaj Ahmad Khan and Khaled Salah. IoT security: Review, blockchain solutions, and open challenges. *Future generation computer systems*, 82:395–411, 2018.

[4] Adeela Waqar, Asad Raza, Haider Abbas, and Muhammad Khurram Khan. A framework for preservation of cloud users' data privacy using dynamic reconstruction of metadata. *Journal of Network and Computer Applications*, 36(1):235–248, 2013.

[5] Jiannan Wei, Tran Viet Xuan Phuong, and Guomin Yang. An efficient privacy preserving message authentication scheme for internet-of-things. *IEEE Transactions on Industrial Informatics*, 17(1):617–626, 2020.

[6] Hao Wang and Yujiao Song. Secure cloud-based EHR system using attribute-based cryptosystem and blockchain. *Journal of medical systems*, 42(8):152, 2018.

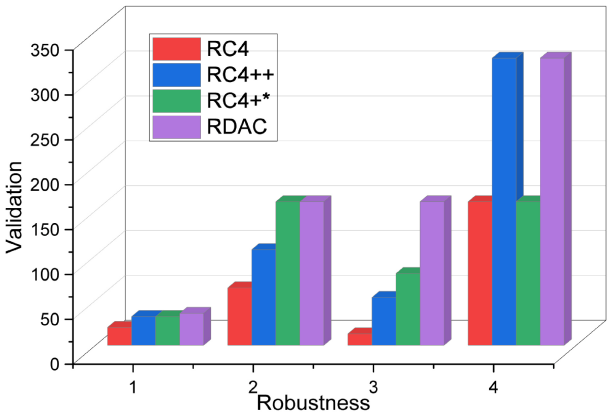


Fig. 7. Robustness of RDAC having s[]=8 byte.

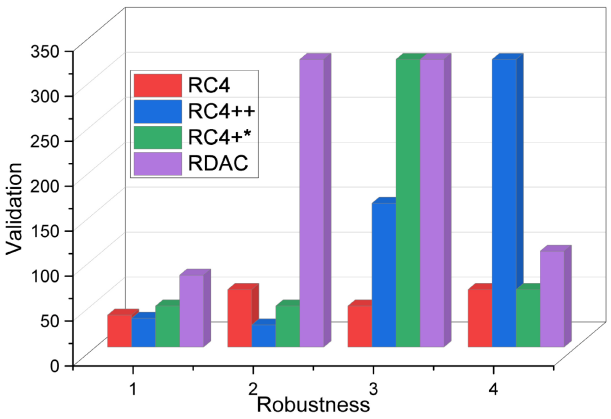


Fig. 8. Robustness of RDAC having s[]=16 byte.

[7] Tiago Guimarães, Ricardo Duarte, João Cunha, Paulo Gomes, and Manuel Filipe Santos. Security and immutability of open data in healthcare. *Procedia Computer Science*, 220:832–837, 2023.

[8] Anca Jurcut, Tiberiu Niculcea, Pasika Ranaweera, and Nhien-An Le-Khac. Security considerations for Internet of Things: A survey. *SN Computer Science*, 1:1–19, 2020.

[9] Jihyeon Ryu, Dongwoo Kang, Hakjun Lee, Hyoungshick Kim, and Dongho Won. A secure and lightweight three-factor-based authentication scheme for smart healthcare systems. *Sensors*, 20(24):7136, 2020.

[10] Seunghwan Son, Yohan Park, and Youngho Park. A secure, lightweight, and anonymous user authentication protocol for IoT environments. *Sustainability*, 13(16):9241, 2021.

[11] Punam Prabha and Kakali Chatterjee. Securing telecare medical information system with blockchain technology. In *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 846–851. IEEE, 2020.

[12] Reza Hooshmand and Mahdi Khoshfekr. Key encapsulation mechanism based on polar codes. *IET Communications*, 16(20):2438–2447, 2022.

[13] Wai-Kong Lee and Seong Oun Hwang. High throughput implementation of post-quantum key encapsulation and decapsulation on GPU for internet of things applications. *IEEE Transactions on Services Computing*, 15(6):3275–3288, 2021.

[14] Mohammad-Hossein Farzam, Siavash Bayat-Sarmadi, and Hatameh Mosanaei-Boorani. Implementation of supersingular isogeny-based Diffie-Hellman and key encapsulation using an efficient scheduling. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(12):4895–4903, 2020.

[15] Sourav Sen Gupta. Analysis and implementation of RC4 stream cipher. *Indian Statistical Institute, India*, 2013.

[16] Sasadhar Bera, Pradeep Kumar, and Subhajit Bhattacharya. A study on how to achieve flexibility in healthcare process: a simulation-based approach. *International Journal of Productivity and Performance Management*, 72(8):2292–2316, 2023.

[17] Sameeh A Jassim and Alaa K Farhan. A survey on stream ciphers for constrained environments. In *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*, pages 228–233. IEEE, 2021.

[18] Serge Mister and Stafford E Tavares. Cryptanalysis of RC4-like ciphers. In *International Workshop on Selected Areas in Cryptography*, pages 131–143. Springer, 1998.

[19] Isnar Sumartono, Andysah Putera Utama Siahaan, and Nova Mayasari. An overview of the RC4 algorithm. *IOSR J. Comput. Eng*, 18(6):67–73, 2016.

[20] Subhadeep Banik, Santanu Sarkar, and Raghu Kacker. Security analysis of the RC4+ stream cipher. In *Progress in Cryptology–INDOCRYPT 2013: 14th International Conference on Cryptology in India, Mumbai, India, December 7-10, 2013. Proceedings 14*, pages 297–307. Springer, 2013.

[21] Anupam Chattopadhyay and Goutam Paul. Exploring security-performance trade-offs during hardware accelerator design of stream cipher RC4. In *2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC)*, pages 251–254. IEEE, 2012.

[22] Subhadeep Banik and Sonu Jha. Some security results of the RC4+ stream cipher. *Security and Communication Networks*, 8(18):4061–4072, 2015.

[23] Robbi Rahim, Iskandar Zulkarnain, and Hendra Jaya. A review: search visualization with Knuth Morris Pratt algorithm. In *IOP Conference Series: Materials Science and Engineering*, volume 237, page 012026. IOP Publishing, 2017.

[24] Kranthi Kumar Mandumula. Knuth-Morris-Pratt. *Indiana State University Terre Haute IN, USA December*, 16, 2011.

[25] Ahmed M Elmisery, Seungmin Rho, and Mohamed Aborizka. A new computing environment for collective privacy protection from constrained healthcare devices to IoT cloud services. *Cluster Computing*, 22:1611–1638, 2019.

[26] Arwa Badhib, Suhair Alshehri, and Asma Cherif. A robust device-to-device continuous authentication protocol for the internet of things. *IEEE Access*, 9:124768–124792, 2021.

[27] Mohammad Wazid, Ashok Kumar Das, Neeraj Kumar, and Athanasios V Vasilakos. Design of secure key management and user authentication scheme for fog computing services. *Future Generation Computer Systems*, 91:475–492, 2019.

[28] Jiangfeng Sun, Fazlullah Khan, Junxia Li, Mohammad Dahman Alshehri, Ryan Alturki, and Mohammad Wedyan. Mutual authentication scheme for the device-to-server communication in the Internet of Medical Things. *IEEE Internet of Things Journal*, 8(21):15663–15671, 2021.

[29] Bakkiam David Deebak, Fadi Al-Turjman, Moayad Aloqaily, and Omar Alfandi. An authentic-based privacy preservation protocol for smart e-healthcare systems in IoT. *IEEE Access*, 7:135632–135649, 2019.

[30] Mohammad Wazid, Ashok Kumar Das, Vivekananda Bhat, and Athanasios V Vasilakos. LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment. *Journal of Network and Computer Applications*, 150:102496, 2020.

[31] B Mamta, B Gupta, KC Li, VCM Leung, KE Psannis, and S Yamaguchi. Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system. IEEE/CAA J Autom Sin, 2021.

[32] Mrs USHARANI Chelladurai, Seethalakshmi Pandian, and Krishnamoorthy Ramasamy. A blockchain based patient centric electronic health record storage and integrity management for e-health systems. *Health Policy and Technology*, 10(4):100513, 2021.

[33] Punam Prabha and Kakali Chatterjee. Design and implementation of hybrid consensus mechanism for IoT based healthcare system security. *International Journal of Information Technology*, 14(3):1381–1396, 2022.

[34] Punam Prabha and Kakali Chatterjee. Rshealth: A ring signature scheme for identity anonymization and transaction privacy in blockchain based e-healthcare systems. *IEEE Access*, 12:117701–117720, 2024.

[35] Punam Prabha and Kakali Chatterjee. Towards a robust data privacy framework for iot based health monitoring system. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, 12:100961, 2025.

[36] S Velmurugan, M Prakash, S Neelakandan, and Eric Ofori Martinson. An efficient secure sharing of electronic health records using iot-based hyperledger blockchain. *International Journal of Intelligent Systems*, 2024, 2024.

[37] D Bala Gayathri and D Sangeetha. An efficient blockchain-based incremental provable data possession for a secure electronic healthcare system. *Biomedical Signal Processing and Control*, 112:108919, 2026.

[38] Ratan Lal, Ahmed Osama Daoud, Ahmed Gouda Mohamed, and Mohamed Nabawy. Blockchain for safety compliance in construction: A comprehensive literature review. *Buildings*, 16(1):143, 2025.

[39] Yang Xin, Lingshuang Kong, Zhi Liu, Chunhua Wang, Hongliang Zhu, Mingcheng Gao, Chensu Zhao, and Xiaoke Xu. Multimodal feature-level fusion for biometrics identification system on IoMT platform. *IEEE Access*, 6:21418–21426, 2018.

[40] Faisal Alsubaei, Abdullah Abuhussein, Vivek Shandilya, and Sajjan Shiva. IoMT-SAF: Internet of Medical Things security assessment framework. *Internet of Things*, 8:100123, 2019.

[41] Mehedi Masud, Gurjot Singh Gaba, Salman Alqahtani, Ghulam Muhammad, Brij B Gupta, Pardeep Kumar, and Ahmed Ghoneim. A lightweight and robust secure key establishment protocol for Internet of Medical Things in COVID-19 patients care. *IEEE Internet of Things Journal*, 8(21):15694–15703, 2020.

[42] Scott Fluhrer, Itsik Mantin, and Adi Shamir. Weaknesses in the key scheduling algorithm of RC4. In *Selected Areas in Cryptography: 8th Annual International Workshop, SAC 2001 Toronto, Ontario, Canada, August 16–17, 2001 Revised Papers 8*, pages 1–24. Springer, 2001.

[43] Donald E Knuth, James H Morris, Jr, and Vaughan R Pratt. Fast pattern matching in strings. *SIAM journal on computing*, 6(2):323–350, 1977.

[44] Punam Prabha and Anupam Sinha. Securing wlan with proposed RC4++ and RC4+* symmetric keystream cipher algorithms. In *2021 2nd International Conference for Emerging Technology (INCET)*, pages 1–5, 2021.