

Received 11 March 2022, accepted 5 June 2022, date of publication 14 November 2022, date of current version 18 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3217230

# A Secure and Resilient Scheme for Telecare Medical Information Systems With Threat Modeling and Formal Verification

SHAIK SHAKEEL AHAMAD<sup>1</sup>, MOHAMMED AL-SHEHRI<sup>1</sup>, AND ISMAIL KESHTA<sup>2</sup>

<sup>1</sup>Department of Information Technology, College of Computer and Information Sciences, Majmaah University, Al-Majmaah 11952, Saudi Arabia


<sup>2</sup>Computer Science and Information Systems Department, College of Applied Sciences, AlMaarefa University, Riyadh 13713, Saudi Arabia

Corresponding author: Shaik Shakeel Ahamad (ahamadss786@gmail.com)

This work was supported by the Researchers Supporting Program, AlMaarefa University, Riyadh, Saudi Arabia, under Grant TUMA-Project-2021-14. The work of Shaik Shakeel Ahamad was supported by the Deanship of Scientific Research at Majmaah University under Project R-2022-295.

**ABSTRACT** Telecare Medical Information Systems (TMIS) is a highly focused and unique domain providing healthcare services remotely, the development and advancement in the realm of information and communication technologies boosted the development of TMIS. Smartphones, IoT devices, Mobile Healthcare Applications (MHA) and hospital servers are the building blocks of TMIS. Emergen Research predicts that IoT based healthcare security market will reach USD 5.52 Billion in 2028. Existing IoT based healthcare solutions are facing many security problems which includes information leakage, false authentication, key loss and are not in compliance with Health Insurance Portability and Accountability Act (HIPAA) regulations as IoT devices and sensors used are prone to Blue Borne, DoS (Denial of Service), DDoS (Distributed Denial of Service) and Reverse-engineering attacks. In addition to these healthcare applications in the IoT devices/sensors and mobile healthcare applications in the smart phone of the patient are vulnerable to repackaging attacks and lacked transport layer protection. This paper proposes a SRSTMIS (Secure and Resilient Scheme for Telecare Medical Information Systems) containing its architecture, a procedure to verify the safety and security of patients credentials and Mobile Healthcare Applications (MHA) and finally proposed a secure protocol. White-Box Cryptography (WBC) ensures the safety and security of the keys in the healthcare applications and in the SE, UICC and TPM. We have threat modeled our proposed healthcare framework using STRIDE approach and successfully verified using Microsoft Threat Modeling tool 2016. Our proposed secure and lightweight authentication scheme has been successfully verified with BAN (Burrows, Abadi, and Needham) logic and Scyther tool, and our proposed protocol overcome DoS (Denial of Service), multi-protocol attack, Blue Borne attack, DDoS (Distributed Denial of Service) attack, reverse engineering, insider, outsider and Phlashing attacks. SRSTMIS overcomes information leakage from sensors during rest and during transit, key loss from healthcare applications of the sensors and smart phone and false authentication and ensures HIPAA regulations. Proposed protocol was successfully implemented in Android Studio. We have compared our proposed work with the existing works and found to better in terms of security, resisting attacks, and in consumption of resources.

**INDEX TERMS** Telecare medical information systems (TMIS), SRSTMIS (secure and resilient scheme for telecare medical information systems), mobile healthcare applications (MHA), white-box cryptography (WBC), health insurance portability and accountability act (HIPAA), BAN logic, and blue borne attack, phlashing attacks, STRIDE approach, scyther tool, microsoft threat modeling tool 2016, reverse-engineering attacks, kotlin language.

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

## I. INTRODUCTION

The rapid advances and development of information and communication technologies boosts the development of Telecare

Medical Information Systems(TMIS) as it is an important domain in modern healthcare which ensures medical services remotely for critically ill patients and elderly people. TMIS is playing a crucial role in the ongoing COVID 19 pandemic as TMIS monitors the patients' health and provides treatment using Smartphones, IoT devices, Mobile Healthcare Applications (MHA) and hospital servers so these are the main building blocks of TMIS. TMIS helps in stopping COVID 19 pandemic's spread as TMIS establishes secure communication among all the entities involved in the TMIS ecosystem. The demand for TMIS is increasing rapidly due to the COVID-19 pandemic. MHAs are extremely helpful in providing an online communication platform reducing physical attendance for unnecessary appointments at the hospitals. Emergen Research predicts that IoT based healthcare security market will reach USD 5.52 Billion in 2028. During COVID-19 pandemic there was a huge rise in the cyberattacks on healthcare systems, while many healthcare systems are inefficient to defend these attacks. Medical data are often very sensitive and needs to be protected. Most of the entities involved in the healthcare framework cannot withstand cyberattacks. Mobile healthcare industry should comply with HIPAA standards in order to regulate data privacy for personal healthcare information. Authors of [27] and [28] proposes Telecare medicine information system (TMIS) frameworks which fail to ensure end to end security. Existing TMIS solutions are facing many security problems which includes information leakage, false authentication, key loss and are not in compliance with Health Insurance Portability and Accountability Act (HIPAA) regulations as sensors used are prone to DoS (Denial of Service), Blue Borne and DDoS (Distributed Denial of Service) attacks. The main targets of Intruders are device, application, data in transit and the data at rest for getting patient's data and credentials (keys) in TMIS. Healthcare applications play vital role in the success of TMIS, but these applications in the sensor and smart phone are not trustworthy as these are prone to reverse engineering attacks from the intruders/attackers. Attackers target these applications for patient's data and credentials (keys) and are often successful in getting patient's data and credentials (keys). In addition to these existing TMIS schemes has more communication and computational cost and are not practical.

### Motivation

- a) **IoT security market:** Emergen Research predicts that IoT based healthcare security market will reach USD 5.52 Billion in 2028.
- b) **Application security market:** According to market watch, the Application Security Market will cross US\$ 11 billion by 2024 globally (Application Security Market) [23].
- c) According to marketsandmarkets IoT medical devices are will reach USD 63.43 billion by 2023 globally (Shelly Singh) [24].
- d) IoT medical devices are being used by many patients all around the globe as they make the life of patients easy and is evident from the predictions

from marketsandmarkets (Application Security Market) [23], but these devices should be made secure right from the manufacturing phase of these devices which is the responsibility of the manufacturer. IoT medical devices use healthcare applications and applications need to be portable and secure, the security of these applications is the responsibility of the hospitals and the government.

- e) Transport layer protection is absent in the existing healthcare applications and are also prone to repackaging attacks.

The remaining article's organization is as follows: In Section II, we present the related work in the realm of TMIS security. In Section III, we propose a SRSTMIS framework. Section IV provides the formal verification SRSTMIS protocol. Section V brings security analysis. Section VI presents threat modeling of SRSTMIS framework. Section VII presents the implementation and performance analysis of SRSTMIS protocol, and Section VIII provides the conclusion of the research work.

## II. RELATED WORK

Reference [2] proposes an authentication scheme in TMIS based on Physical Unclonable Function (PUF) and Elliptic Curve Cryptography (ECC) technology. But this solution has no clarity

- a) How the ECC technology can encrypt the messages in the real time.
- b) How the healthcare application overcomes reverse engineering attacks?

Reference [3] proposes healthcare systems which ensures privacy location, mutual authentication and with less storage and computational costs, but the application and communication security was compromised. Reference [4] proposes an authentication scheme for IoT healthcare based on cloud, but it suffers from repackaging attacks and lacked transport layer protection. Dhillon and Kalra [6] uses ECC (Elliptic Curve Cryptography) algorithm for proposing an authentication scheme for healthcare which monitors patients remotely. The main limitation of this work is medical professional or doctor will be able to access patient's data. Sharma and Kalra [4] proposed user authentication scheme which is lightweight in healthcare based on Cloud. The main contribution of this work is hospital can get the real-time data from the sensor of a remote patient and this data can be stored in the cloud server. But this work has the following limitations listed below

- a) Data security in the cloud is not ensured
- b) This work does not ensure non-repudiation and accountability properties

A novel authentication scheme is proposed by Kumar et al. [7] in the realm of Wireless Medical Sensor Networks, but we have found that the proposed work is very much vulnerable to insider attack, does not ensure end to end security and is prone to off-line password guessing attack.

The research work proposed by Li et al. [8] cannot withstand impersonation and off-line password guessing attacks.

The work proposed by Wu et al. [9] is prone to multi-protocol, Blue Borne, DDOS, reverse engineering and Phlashing attacks. Salem and Amin [10] proposed a privacy protection protocol based on the El-Gamal cryptographic system to improve the medication security of patients in TMIS. However, the storage cost of this protocol is too high. Xu et al. [11] proposed a PUF-based lightweight RFID security protocol to achieve effective verification of a single tag but this work is vulnerable to desynchronization attacks and secret disclosure attacks. Most of the works discussed in this section use RFID technology with RFID tags and these are the main targets of attackers. Following are the limitations of using RFID in TMIS

- a) There is no clarity how the credentials are stored in RFID tags
- b) There is no clarity how the credentials are stored in the gateway if they are stored in the memory of the gateway they are prone to attacks.
- c) RFID tags can be reverse engineered and there is no mechanism to overcome these (reverse engineering) attacks
- d) There is no safety and security of keys in RFID tags
- e) RFID tags are prone to information leakage
- f) RFID tags are prone to false authentication attack

Reference [12] proposes a Tele-COVID application which is both web and Android based telemedicine application monitoring COVID patients. Following are the drawbacks of this research work

- a) There is no mention how the credentials are generated and stored in the Tele-COVID application
- b) Tele-COVID application is prone to reverse engineering attack.
- c) Application security and Communication security is not ensured
- d) This work is not in compliance with HIPAA regulations

Reference [22] proposes a block chain-based healthcare system sharing with cloud-based services the main contribution is Access control mechanism and the drawbacks are Scalability and key management. Following are the drawbacks of block chain based healthcare systems

- a) The size of Block chain ledger will increase as the time passes which makes the record management difficult for IoT devices such as sensors
- b) Sensors are resource constrained devices and block chain uses asymmetric encryption algorithms which require more processing power and time thereby consuming more battery.
- c) There is no central database in block chain to store patient's information so the ledger needs to be stored on the participating sensors as the block chain increases in size which makes it difficult for sensors as they have very less storage capacity.
- d) Block chain depends heavily on private key if the private key is compromised or lost all the patient's information is lost and moreover when the patient

is unconscious doctors cannot retrieve his medical records without patient's private key.

- e) Immutability property of a block chain hinders the adoption of block chain technology in healthcare as patient cannot erase his own health information/records.
- f) Block chain based healthcare solutions introduce latency
- g) Block chain based healthcare solutions are not practical to store high-volume of healthcare information or data on block chain as this is will degrade the performance.

So we haven't adopted block chain technology and proposed this research work which ensures defense in depth security and is resilient, so security is ensured in all the entities involved in the ecosystem and at all the levels. Security is incorporated in the design phase and implementation phase of our proposed healthcare system.

Following are the main limitations in the existing literature

- a) Mutual authentication between the IoT Sensor (SUCH AS WHRM) and hospital is not ensured.
- b) Existing TMIS schemes are prone to information leakage, key loss and false authentication.
- c) Healthcare applications are prone to reverse engineering attacks.
- d) Existing IoT based healthcare schemes are prone to IoT device specific attacks which includes BlueBorne, DDOS attacks in IoT based healthcare.
- e) Most of the existing works are not practical
- f) Very few solutions/schemes in the existing literature were implemented in the real time.
- g) The communication cost and computational cost of the existing TMIS schemes are more.
- h) Existing TMIS schemes does not comply with HIPAA regulations.

So there is a great need of secure and resilient scheme in TMIS. All the entities involved in the framework should be able to withstand, avoid and recover from attacks targeting patient's keys and confidential data in SE, UICC, TPM and MHA. TMIS framework should ensure security and safety at the 'device level', 'application level', during transit and at rest.

**Novelty of our research work:** The novelty of our proposed work are:

- a) As per our knowledge we are the first to address key loss, false authentication and information leakage issues in the realm of TMIS.
- b) As per our knowledge our proposed work is the only work which ensures the safety and security of keys in IoT sensors, smartphones and healthcare applications.
- c) As per our knowledge our proposed work is the only work in TMIS which overcomes reverse engineering attacks on Healthcare applications.
- d) As per our knowledge we are the first to address BlueBorne, DDOS attacks in IoT based TMIS schemes.
- e) As per our knowledge we are the first to threat model our proposed SRSTMIS framework using STRIDE

approach and successfully verified using Microsoft Threat Modeling tool 2016.

- f) Our proposed SRSTMIS framework is resilient as the intruder/attacker will not be successful in extracting and manipulating the credentials from any of the devices involved in the SRSTMIS framework. In addition to this SRSTMIS framework ensures the security of data at the device level, application level, at rest and during the transit.

**Contributions made:** The contribution made by this work are as follows:

- a) Proposes a TMIS architecture, a procedure to verify the safety and security of patients credentials and Mobile Healthcare Applications (MHA) and finally proposed a secure protocol.
- b) Proposed healthcare scheme overcomes information leakage, key loss and false authentication.
- c) We have threat modeled our proposed healthcare framework using STRIDE approach and successfully verified using Microsoft Threat Modeling tool 2016.
- d) This research work overcomes information leakage from sensors during rest and during transit, key loss from healthcare applications of the sensors and smart phone and false authentication among the entities involved in the system thereby ensuring HIPAA regulations.
- e) Proposed secure TMIS scheme ensures confidentiality, integrity, availability, mutual authentication and non-repudiation properties.
- f) Proposed secure TMIS scheme overcomes multi-protocol attack, Blue Borne, DoS, DDoS, reverse engineering and Phlashing attacks.
- g) SRSTMIS's energy, communication, and computation costs are far less than that of the existing TMIS research works.
- h) SRSTMIS is formally verified using BAN logic [17] and [18], and Scyther tool [19] and [20].
- i) We have successfully implemented our proposed SRSTMIS in Android Studio.

Motivated by these solutions, we find no work till date which ensures the safety and security of keys and healthcare applications both in sensor and in patient's smartphone. Existing solutions are prone to information leakage, false authentication and vulnerable to repackaging attacks. We name our proposed framework as Secure and Resilient Authentication Scheme in TMIS (SRSTMIS).

### III. PROPOSED SECURE TMIS FRAMEWORK

#### A. THREAT MODEL AND OUR PROPOSED TMIS ARCHITECTURE

##### 1) THREAT MODEL

The Dolev and Yao [25] model explains about an attacker's capability in between two parties communicating each other through a channel which is open. Following are the capabilities of an attacker:

- (i) An attacker has the capability to access the data stored in the memory of Sensor/Smartphone/Hospital Server.
- (ii) An adversary has the capability to tamper the patient's data and credentials in the ecosystem.
- iii Attackers has the capabilities to replay, update remove the data exchanges in the ecosystem.
- (iv) An attacker can also access the credentials and data sensor/smartphone of a doctor by reverse engineering.

##### 2) PROPOSED TMIS ARCHITECTURE

SRSTMIS's main entities are Certification Authority (CA), Patient's smart phone (P), Sensor i.e. WHRM (S), Hospital (H), Doctor (D), MHA Manufacturer (MM), Sensor Manufacturer (SM). CA and Hospital (H) has a TPM. BLE technology is used in the proposed TMIS framework. Patient (P), Patient's smart phone (P), Sensor Node (S), Hospital Server (H), and Doctor will be the first to register with the CA. Sensors (like WHRM) are on the patient's body which can sense the working of WHRM and broadcasts the readings to the Patient's smart phone (P) through resource constraint sensor (S). Patient's smart phone (P) transfers data to doctor (D) and the Hospital Server (H) for possible diagnosis. The SRSTMIS architecture is shown in Figure 1.

- a) **Sensor (S):** It is an IoT device such as a Wearable Heart Rate Monitor (WHRM), with a processing unit, eight-bit microcontroller and a communicates with a Bluetooth Low Energy (BLE). 'S' is a low power device operates with a coin cell battery. 'S' implements Trusted Computing Base (TCB) which enables a MHA to run reliably, securely, and with high quality. Sensor (S) collects patient's readings and forwards it to the UICC of the patient's smartphone at regular intervals via Bluetooth Low Energy (BLE); in order to overcome BLE vulnerabilities, MHA in Sensor (S) encrypts the data sent to the MHA in the UICC of the smartphone (patient (P)). Patient's readings are
  - b) encrypted using the shared symmetric key between the MHA of the Sensor (S) and MHA in UICC (P).
  - c) **Patient (P):** Hospital is a registered entity possessing a smartphone with MHA.
  - d) **MHA:** Mobile Healthcare Application (MHA) is an entity which interacts or communicates with other entities. MHA is installed in 'S', 'UICC', 'H' and 'D'.
  - e) **Hospital (H):** Hospital is an entity which provides treatment for its patients, it has a server with TPM in addition to database. TPA (Third Party Auditor) is employed by CA as CA is a TSM (Trusted Service Manager) in SRSTMIS. TPA monitors and audits all the activities in the hospital which includes Logging and Monitoring. Hospital premises has Hospital Database, Hospital TPM, TPA and is protected by Private Hospital Network (PHN), PHN is a dedicated network which connects differed sub-entities in the hospital premises, outside traffic is not allowed in the PHN. Messages are exchanged among the entities in the PHN are protected using IPSec protocol. SRST-

TABLE 1. List of abbreviations and notations.

Abbreviation	Description
BAN	Burrows, Abadi, and Needham
DOS	Denial Of Service
DDOS	Distributed Denial Of Service
MHA	Mobile Healthcare Application
AES	Advanced Encryption Standard
GCM	Galois/Counter Mode
UICC	Universal Integrated Circuit Card
HSM	Hardware Security Module
TSM	Trusted Service Manager
CA	Certifying Authority
MPKI	Mobile Public Key Infrastructure
HIPAA	Health Insurance Portability and Accountability Act
COS	Card Operating System
SE	Secure Element
TA	Trusted Authority
OTA	Over The Air
MNO	Mobile Network Operator
WBC	White Box Cryptography
EAL 4+	Evaluation Assurance Level 4+
TLS	Transport Layer Security
OCSP	Online Certificate Status Protocol
TEE	Trusted Execution Environment
SPDL	Security Protocol Description Language
TAC	Traceable anonymous certificate
P	Patient's smart phone
S	Sensor i.e. WHRM
H	Hospital
D	Doctor
MM	MHA Manufacturer
SM	Sensor Manufacturer
BLE	Bluetooth Low Energy
RA	Registration Authority
TPM	Trusted Platform Module
TMIS	Telecare Medical Information Systems
IOT	Internet Of Things
PUF	Physical Unclonable Function
RFID	Radio Frequency Identification
TSM	Trusted Service Manager
APK	Android Package Kit or Android Application Package
ECDH	Elliptic Curve Diffie Hellman
RBAC	Role-Based Access Control
ID <sub>P</sub>	Patient's Identity
ID <sub>S</sub>	Sensor's Identity
T <sub>S</sub>	Timestamp of Sensor
N <sub>S</sub>	Nonce of Sensor
LOC <sub>P</sub>	Location of Patient
PD	Patient's Data
SK <sub>PS</sub>	Shared Symmetric Key between Patient & Sensor
SK <sub>PH</sub>	Shared Symmetric Key between Patient & Hospital
ID <sub>H</sub>	Hospital's Identity
T <sub>P</sub>	Timestamp of Patient
N <sub>P</sub>	Nonce of the Patient
T <sub>H</sub>	Timestamp of Hospital
N <sub>H</sub>	Nonce of the Hospital
SK <sub>HD</sub>	Shared Symmetric Key between Hospital & Doctor
WHRM	Wearable Heart Rate Monitor
PU <sub>E</sub>	Public key of the entity of SRSTMIS
PR <sub>E</sub>	Private key of the entity of SRSTMIS

MIS framework resists DoS and DDoS attacks as 'H' detects these attacks by change-point detection, activity profiling and wavelet-based signal analysis detection techniques. In addition to these 'H' installs DoS/DDoS protection tools such as "Fort Guard Anti-DDoS". In order to collect evidence from 'H', TPA collects the evidence from its networks, firewalls, IDPS (Intrusion Detection and Prevention System) and hospital TPM. In addition to these TPA gets vital evidence of DoS attack attempts from the logs of "Fort Guard Anti-DDoS" tool.

- f) **TPM:** Both the Hospital (H) and CA use TPM. TPM adds security and integrity for the Hospital and CA's servers as it protects their credentials such as keys, tokens and ensures the integrity of hardware platforms and host Operating Systems. Hardware controller on the server's motherboard is implemented by TPM of the 'H' and 'CA', so hardware controller acts as a repository for securely storing the credentials which includes passwords, tokens, keys, and digital certificates. TPM is immune to malware and forgery. TPM creates a "fingerprint" of the server with its components as it boots, and comparing that baseline against periodic measurements of the system's parameters if there is any deviation it indicates that the server was compromised and the server will not boot. If the hospital/CA server boots successfully with the TPM then the server is not compromised and it can be trusted. MPKI is implemented in TPM. The combination of MPKI, WBC and TPM makes the Hospital (H) and CA servers immune to attacks.
- g) **MHA Manufacturer (MM):** MM manufactures MHA needed in the framework. MM manufactures and distributes MHA to the hospitals and is responsible for the security of the MHAs.
- h) **Sensor Manufacturer (SM):** "SM" manufactures IoT medical devices such as WHRMs and cardiac resynchronization therapy (CRT) devices, which are very much helpful in monitoring the patients remotely and improves the efficiency in addition to reducing the cost. These devices help in monitoring and in assessing the condition of patients, it has bidirectional communication features. SM embeds Secure Element (SE) in the Sensor (S), Hospital installs MHA in the 'S'. White Box Cryptography (WBC) and MPKI are implemented in the Secure Element (SE) of these devices (i.e. S) which help in the safety and security of keys and MHA.
- i) **Mobile Public Key Infrastructure (MPKI):** In order to ensure all the security properties in the proposed TMIS we need to adopt MPKI, but the implementation of MPKI in the memory of Sensor (S), TPM and smart-phone is suicidal as the keys can be compromised. TPM and UICC generates and stores their credentials.
- j) **UICC and SE:** A Secure Element (SE) and UICC are tamper-resistant hardware devices with the capability to host mobile applications. UICC hosts different mobile applications in separate security domains which is controlled by the Owner of the application. UICC implements firewalls among applications which restricts mobile applications from interfering. Patient's anonymity in our proposed SRSTMIS framework is ensured using TAC (Traceable anonymous certificate) [26], UICC and SE implements MPKI and WBC which helps in ensuring the safety and security of keys and healthcare applications.
- k) **Certifying Authority (CA):** CA plays the role of an adjudicator and Trusted Entity (TE)/ Trusted Service

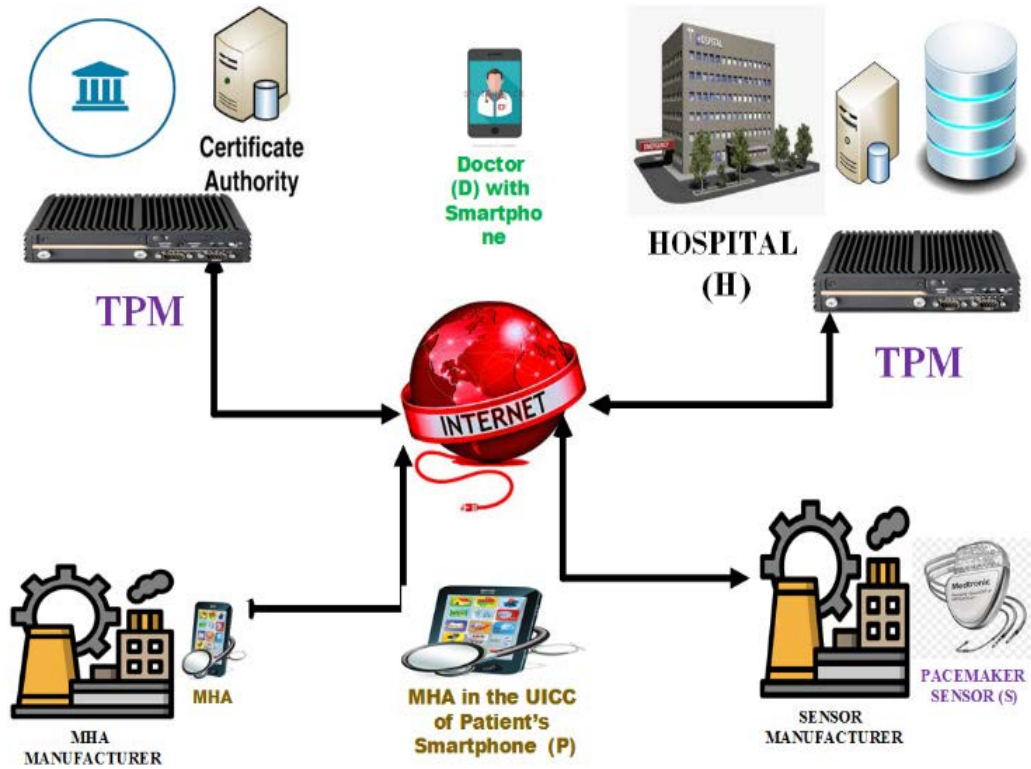


FIGURE 1. Proposed TMIS Architecture.

Manager (TSM) and is trusted by all the participants in the framework or ecosystem, it issues and revokes certificates to all the participants. Registration Authority (RA) works under the supervision of CA, RA’s responsibility includes verifying the credentials of the entities’ involved in the ecosystem. In our proposed framework Hospital (H) acts as a RA. OCSP (Online Certificate Status Protocol) is a subsidiary of CA, revokes compromised certificates. CA is the Root of Trust (ROT) in our proposed framework which decides the cryptographic procedures and policies that helps in governing how communications, applications and identities should be secured using these cryptographic procedures and policies

**B. PROPOSED PROCEDURE TO VERIFY THE SAFETY AND SECURITY OF CREDENTIALS AND MOBILE HEALTHCARE APPLICATION (MHA)**

Our proposed work ensures Application security, Endpoint security and Network security. Self-Signing Restriction, Code Attestation, Control Flow Obfuscation. The main difference between Code obfuscation and WBC, Code obfuscation hides the complete variables, program, flow, but in WBC key is private which is a secret and the algorithm is public so the parameters related to the key cannot be retrieved by an attacker who is in possession of the Medical sensor and smart

phone. Our proposed framework overcomes BLE vulnerabilities as our MHA’s code is obfuscated by the MHA manufacturer and attested by the Certifying Authority (CA) and imposes self-signing restrictions, in addition to these Sensor (S) transmits encrypted data using the symmetric key shared between sensor’s MHA and the MHA of the patient (P). Data encryption prevents MITM and eavesdropping attacks. A secure link is established between the sensor’s MHA and MHA in the UICC of the patient ensuring application security (symmetric key) and communication security (using SSL/TLS).

MM manufactures and distributes MHA to the hospitals and is responsible for the security of the MHAs. In the process of securing the MHAs from reverse engineering attacks which is one of the dangerous attacks against MHA, MM implements the following countermeasures:

- i) **Logic Obfuscation:** Our proposed framework adopts logic Obfuscation which prevents the attackers to know the logic of the healthcare application.
- ii) **Control Flow Obfuscation:** Our proposed framework adopts control flow obfuscation, MM reorganizes the control flow of the MHA, injects dummy code, removes functions’ makes use of proxy methods to redirect the flow of execution and the process tree.
- iii) **Self-Signing Restriction:** MHAs are digitally signed only by CA; no other entity has the authority to sign an a healthcare application.

## 1) VERIFICATION OF REVERSE ENGINEERING ATTACK ON MHA BY THE HOSPITAL (H)

Hospital verifies the reverse engineering attack on MHA as given in algorithm 1. Hospital (H) acquires the package file (.apk) of the MHA and starts analyzing APK file of the MHA using jadx-gui, opens and analyzes MHA's manifest file i.e. AndroidManifest.xml, if it finds android:debuggable="true" and android:allowBackup="true" then the hospital comes to the conclusion that MHA was compromised or else MHA was not compromised. Hospital Test the parameters/flags in WebViews if *setWebContentsDebuggingEnabled(true)* and *clearCache(true)* are enabled to true then the hospital concludes that the MHA was compromised.

---

### Algorithm 1 Reverse Engineering Attack Verification on MHA by Hospital (H)

---

**Input:** Executable File of MHA

**Output:** Result whether MHA was reverse engineered or not

**Step 1: Hospital (H)** acquires package file (.apk) of the MHA

**Step 2:** Analyze APK file of MHA using jadx-gui

**Step 3:** Analyze MHA's manifest file i.e. AndroidManifest.xml

```
If
    android:debuggable = "true"
    android:allowBackup = "true"
```

```
Then
    MHA is compromised
```

```
Else
    MHA is not compromised
```

**Step 4:** Test the parameters/flags in WebViews

```
If
    setWebContentsDebuggingEnabled(true)
    clearCache(true)
```

```
Then
    MHA is compromised
```

```
Else
    MHA is not compromised
```

```
Exit
```

---

## 2) VERIFICATION OF MHA BY THE HOSPITAL

Hospital verifies the MHA as given in algorithm 2. Hospital (H) first verifies the digital signature (which was generated by CA) on the MHA, if the verification was successful, it then moves to step 2. Step 2 verifies whether the MHA is Obfuscated or not. H gets a file (.apk) of the MHA which involves analyzing APK file of MHA using jadx-gui, checks whether the code is obfuscated, if it finds the code to be obfuscated then it concludes that the MHA cannot be compromised, then it moves to step 3. Hospital (H) verifies whether the MHA was Reverse Engineered or not by using the algorithm 1. If all the three steps are successful, then the Hospital (H) that the MHA was not compromised.

### C. PROPOSED SCHEME

Figure 1 shows an e-healthcare architecture using TMIS, which consists of four types of entities such as Patient (P), Hospital (H), cloud server (C), Certifying Authority (CA),

---

### Algorithm 2 MHA Verification by the Hospital

---

**Input:** MHA to be verified

**Output:** Result whether MHA was fabricated/compromised or not **Step 1:** Hospital (H) verifies whether the MHA is digitally signed by the CA or not

```
If
    Verification of Digital Signature generated by the CA "="
    TRUE"
```

```
Then
    MHA was not Tampered
```

```
Else
    MHA was Tampered
```

**Step 2:** Hospital (H) verifies whether the MHA is Obfuscated or not. H gets a file (.apk) of the MHA.

Step 2.1: Analyzes APK file of MHA using jadx-gui

Step 2.2: Verifies the logic of the code

```
If
    MHA code was Obfuscated logically = " TRUE"
```

```
Then
    MHA cannot be Tampered/Compromised
```

```
Else
    MHA can be Tampered /Compromised
```

**Step 3:** Hospital (H) verifies whether the MHA was Reverse Engineered or not.

*Hospital (H) verifies the MHA using the Algorithm 1*

If any of the THREE Steps fails, the Hospital (H) comes to the conclusion that MHA was compromised.

*Exit*

---

Application Provider (AP), IoT Medical Sensor Manufacturer (M) and Doctor (D). CA issues X.509 certificates to all the participants in the ecosystem, and all the participants have their own key pairs and all the participant's certificates are in the CA's directory. Patient and Doctor possesses UICC in his/her smartphone, IoT sensor secure element, Mobile Healthcare Application (MHA) is in the UICC of the Patient and Doctor's smartphone, MHA is also a part of secure element in IoT sensor. BLE (Bluetooth Low Energy) is a communication technology used between the IoT sensor and patient's smartphone.

#### SRSTMIS Phases

Our proposed SRSTMIS protocol comprises the following phases

- Setup, Trusted Storage and Key Management Phase in SRSTMIS
- Registration and Key Agreement phase between H and IoT Medical Sensor (S)
- Registration and Key Agreement phase between H and Patient (P)
- Registration and Key Agreement phase between H and Doctor (D)
- Health Monitoring Phase

We have provided all the required notations in Table 1 and illustrate the above phases as follows.

- Setup, Trusted Storage and Key Management Phase in SRSTMIS:** All the participants involved in the 'SRSTMIS' framework except 'S' generate their MPKI credentials in tamper resistant hardware devices, 'H' generates its credentials in 'TPM', 'P' and 'D'

generates their credentials in ‘UICC’ of their smartphones. All the participants generate their credentials which involves a public and private/secret keys. CA verifies the possession of private key for an equivalent public key, after successful verification of private key CA issues a certificate for that participant. All the entities in SRSTMIS has trusted storage, which helps them to securely store their credentials and MHA thereby ensuring the integrity and confidentiality of the credentials and MHA. SRSTMIS framework never allows to export the credentials and MHA to other entities without proper mutual authentication and authorization, in addition to these SRSTMIS allows only ‘H’ to update the MHA OTA (Over The Air) using a secure tunnel at regular intervals. Security of the keys and MHA relies on the tamper resistance nature of the SE, UICC and TPM and the WBC (White-Box Cryptography), as these devices (SE, UICC and TPM) can securely store keys, generate random numbers, encrypt messages (using both symmetric and asymmetric), perform hashing and implements WBC (White-Box Cryptography).

**b) Registration and Key Agreement phase between H and IoT Medical Sensor (S):** Hospital buys IoT Medical Sensors (S) from the IoT Sensor Manufacturer (M), ‘S’ contains a Secure Element (SE), MPKI credentials are not installed in ‘S’, ‘S’ can be a WHRM. ‘H’ installs MHA in the SE of ‘S’, generates and installs the shared symmetric key (AES -256) shared between ‘H’ and ‘S’ in the MHA of ‘S’ physically. Installed MHA and it’s shared symmetric key cannot be extracted or tampered as it is protected by WBC and moreover MHA cannot be reverse engineered as it is code and control obfuscated and signed by the CA. Sensors (S) cannot be personalized/customized OTA (Over The Air) by ‘H’, as it is not connected to ‘H’ directly, it is connected through ‘P’. ‘S’ is a secure element which is a tamper-resistant security device which can implement AES and SHA-256 algorithms. In addition to AES and SHA-256 algorithms it has a Random Number Generator (RNG) which generates random numbers (RN). ‘S’ generates a session key  $SeK_{HS}$  from the  $SK_{HS}$  (symmetric key) which is embedded in the SE of ‘S’, in order to generate a session key which is valid for only one session, ‘S’ uses RNG for generating RN,  $SK_{HS} + RN = SeK_{HS}$ , ‘S’ encrypts the message with the session key  $SeK_{HS}$  and sends the encrypted message along with ‘RN’ to the ‘H’. ‘H’ will generate  $SeK_{HS}$  from the  $SK_{HS}$  and ‘RN’ as follows

$$SK_{HS} + RN = SeK_{HS}$$

After generating the session key  $SeK_{HS}$  ‘H’ decrypts the received message, so this process completely eliminates the possibility of compromising the session keys which are in transit. White Box Cryptography

(WBC) [21] ensures the secure storage of shared symmetric and session keys in the MHA of ‘S’. Figure 2 shows the Registration and Key Agreement Phase between ‘H’ and ‘S’.

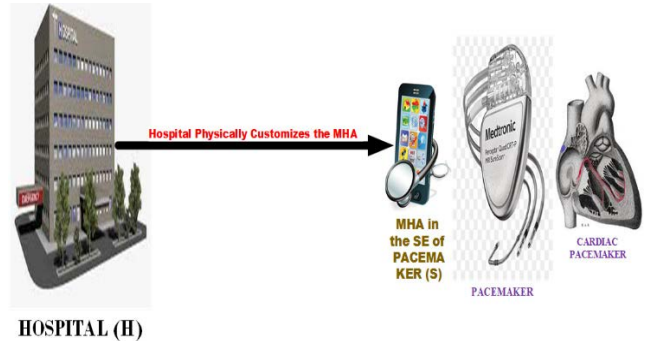


FIGURE 2. Registration and Key Agreement Phase between ‘H’ and ‘S’.

**c) Registration and Key Agreement phase between H and Patient (P):** Patient visits the hospital physically and submits his National ID, Mobile number, Digital Certificate (signed by the CA) and his biometric to the hospital, then the hospital generates Bio-Hash of the biometric and keeps National ID, Mobile number, Digital Certificate and Bio-Hash of the biometric and allocates a patient ID ( $ID_P$ ) to the patient. Hospital allocates username and temporary password to the patient, requests the patients to download the MHA from the hospital URL, ‘P’ downloads, installs and logs in the MHA using the credentials supplied by the hospital and has the provision to change the password. Hospital takes the responsibility of generating and installing the shared symmetric key (AES –256) to the Patient (P), ‘H’ customizes the MHA which is in ‘P’ with symmetric key shared between ‘P’ and ‘H’ Over The Air (OTA). Figure 3 shows the Registration and Key Agreement Phase between ‘H’ and ‘P’

$$\rightarrow HP : \{MS1\}PU_P$$

$$MS1 = \{ID_P, ID_S, SK_{PH}, T_H, N_H\}$$

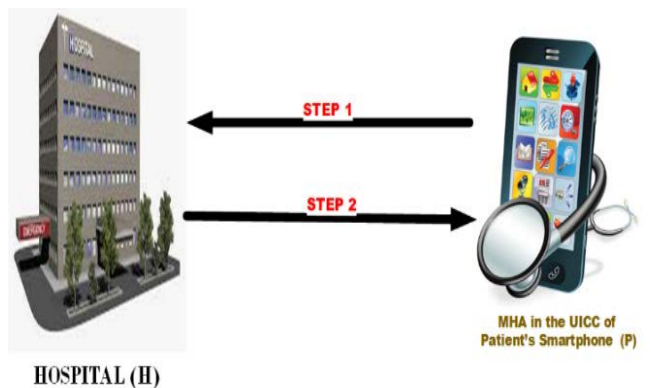


FIGURE 3. Registration and Key Agreement Phase between ‘H’ and ‘P’.



**d) Registration and Key Agreement phase between H and Doctor (D):** After appointing ‘D’ as a Doctor, ‘D’ needs to submit his National ID, Mobile number, Digital Certificate (signed by the CA) and his biometric to the hospital, then the hospital generates Bio-Hash of the biometric and keeps National ID, Mobile number, Digital Certificate and Bio-Hash of the biometric and allocates a Doctor ID ( $ID_D$ ) to the doctor. Hospital allocates username and temporary password to the doctor, requests the doctor to download the MHA from the hospital URL, ‘D’ downloads, installs and logs in the MHA using the credentials supplied by the hospital and has the provision to change the password. Hospital takes the responsibility of generating and installing the shared symmetric key (AES –256) to the ‘D’, ‘H’ customizes the MHA which is in ‘H’ with symmetric key shared between ‘D’ and ‘H’ Over The Air (OTA). Figure 4 shows the Registration and Key Agreement Phase between ‘H’ and ‘D’

$$\begin{aligned} &\rightarrow HD : \{MS2\}PU_D \\ MS2 &= \{ID_D, ID_H, SK_{DH}, T_H, N_H \end{aligned}$$

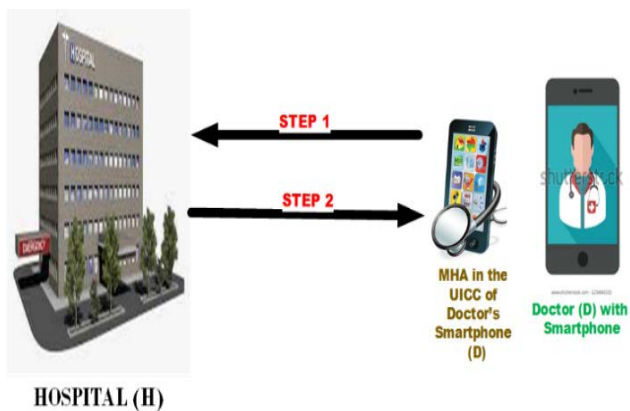


FIGURE 4. Registration and Key Agreement Phase between ‘H’ and ‘D’.

**e) Health Monitoring Phase:** Figure 5 depicts the steps of health monitoring phase.

**Step 1:** ‘S’ collects the patient’s readings at regular intervals and forwards it to the patient’s smartphone (P); ‘S’ communicates with ‘P’ using BLE technology, ‘S’ encrypts the readings with the session key shared between ‘S’ and ‘H’, so ‘P’ will not be able to decrypt the readings, where ‘PR’ is Patients Readings & Random Number ( $RN_S$ ) generated by the Random Number Generator (RNG)

$$\rightarrow SP : \{MS3\}SeK_{HS}, RN_S$$

$$MS3 : \{ID_P, ID_S, T_S, N_S, LOC_S, PR, H(PR)\}$$

**Step 2:** ‘P’ receives the message  $\{MS3\}SK_{HS}$  and makes ‘MS4’ as ‘P’ cannot decrypt the received message as it is encrypted with the symmetric key shared between ‘S’ and ‘H’

$$\rightarrow PH : \{MS5\}SK_{HP}$$

$$MS5 : \{ \{MS3\} SeK_{HS}, RN_S, \{MS4\}$$

$$\{MS4\} = ID_P, ID_S, T_P, N_P, LOC_P$$

**Step 3:** ‘H’ receives the message  $MS5\}SK_{HP}$  from ‘P’ and decrypts the received message using its symmetric key shared with ‘P’. After decrypting ‘MS3’ it gets RN using the received  $RN_S$  and a symmetric key shared between ‘S’ and ‘H’, ‘H’ generates a session key which is used to decrypt the  $MS3\}SeK_{HS}$ . ‘H’ compares the following attributes/fields in ‘MS3’ and ‘MS4’, if the attributes/fields are successfully verified then it checks the PR (Patients Readings).

$$ID_P \text{ (fromMS3)}? = ID_P \text{ (fromMS4)}$$

$$ID_S \text{ (fromMS3)}? = ID_S \text{ (fromMS4)}$$

$$T_S \text{ (fromMS3)}? = T_P \text{ (fromMS4)}$$

$$LOC_S \text{ (fromMS3)}? = LOC_P \text{ (fromMS4)}$$

If the PR (Patients Readings) are within the limits, Hospital (H) updates the database or if there is any deviation, it sends the message to the doctor (D) and patient’s family members. Hospital (H) sends an ambulance to the patient’s location. It also informs to the family members in case of emergency.

$$\rightarrow HD : \{ID_P, ID_H, T_H, N_H, LOC_P, PR\}SK_{HD}$$

**Resilience of ‘SRSTMIS’:** Our proposed framework is resilient as the entities/participants has the ability to withstand, avoid and recover from attacks to compromise the keys, confidential patient’s data in SE, UICC, TPM and MHA in addition to security of confidential patient’s data during the transit thereby ensuring HIPAA regulations. In our proposed ‘SRSTMIS’ TMIS framework security and safety is at levels i.e. ‘device level’, ‘application level’, during transit and at rest. We have used AES-256 algorithm for encrypting the messages exchanged among the participants.

#### IV. FORMAL VERIFICATION

##### A. FORMAL VERIFICATION OF THE PROTOCOL

###### 1) BAN LOGIC PROOF

BAN logic [16], [17], [18] contains many objects classified as principals, keys (symmetric, asymmetric keys and digital signature keys) and statements. These are represented symbolically as

$K_{sh}$ ,  $K_{ph}$  and  $K_{dh}$  are the shared symmetric keys in the SRSTMIS framework. MPKI is adopted in SRSTMIS

framework so  $K_S, K_P, K_H$  and  $K_D$  represents the public keys of ‘S’, ‘P’, ‘H’ and ‘D’ and  $K_S^{-1}, K_P^{-1}, K_H^{-1}$  and  $K_D^{-1}$  represents the private keys of ‘S’, ‘P’, ‘H’ and ‘D’.  $N_S, N_P, N_H$  and  $N_D$  represents the Nonce generated by ‘S’, ‘P’, ‘H’ and ‘D’ participants and finally  $T_S, T_P, T_H$  and  $T_D$  represents the Timestamps generated by ‘S’, ‘P’, ‘H’ and ‘D’ participants in SRSTMIS framework.

BAN LOGIC Constructs

a) **‘E’ believes ‘X’:** When an entity is convinced of the truth of a formula, we say that the entity believes it. Some of these beliefs are introduced as assumptions; the others are concluded in the logic using the predicates.



FIGURE 5. Health Monitoring Phase.

- b) **'E' sees X**: The principal 'E' receives a message containing X. 'E' will decrypt the received message.
- c) **'E' said X**: The principal 'E' believed X when it sent the message
- d) **'E' controls X**: 'E' has jurisdiction over X. The principal 'E' is an authority on X and should be trusted on this matter.
- e) **fresh(X)**: This means that "X" is fresh when 'X' has not been sent in a message at any time before.
- f)  $X \leftrightarrow Y$ : X and Y is a shared symmetric key 'K', both 'X' and 'Y' trust 'K'.
- g)  $\{X\}_K$ : Key 'K' is used to encrypt 'X'

- f) **AS6.T<sub>S</sub>** is timestamp generated by S ensuring timeliness.
- g) **AS7.T<sub>P</sub>** is timestamp generated by P ensuring timeliness.
- h) **AS8.T<sub>H</sub>** is timestamp generated by H ensuring timeliness.

3) SRSTMIS PROTOCOL VERIFICATION USING BAN LOGIC

STEP1 :  $S \rightarrow P : \{MS3\}SeK_{HS}, RN_S$   
 $MS3 : \{ID_P, ID_S, T_S, N_S, LOC_S, PR, H(PR)\}$   
 STEP2 :  $P \rightarrow H : \{MS5\}SK_{HP}$   
 $MS5 : \{ \{MS3\} SeK_{HS}, RN_S, \{MS4\} \}$   
 $\{MS4\} = ID_P, ID_S, T_P, N_P, LOC_P$   
 STEP3 :  $H \rightarrow D : \{ID_P, ID_H, T_H, N_H, LOC_P, PR\}SK_{HD}$

**Step 1**: 'P' receives  $\{MS3\}SeK_{HS}, RN_S$  from 'S' and 'P' cannot decrypt the received message as it is encrypted with the session key shared between 'S' and 'H'. In our proposed 'SRSTMIS' framework a secure tunnel is established among the participants and 'S' can only communicate with 'P'.

**P believes**:  $\{MS3\}SeK_{HS}, RN_S$  — (1)  
**P believesSsaid**:  $\{MS3\}SeK_{HS}, RN_S$  — (2)  
 So, from the statements (1) to (2)  
**P believes**:  $\{MS3\}SeK_{HS}, RN_S$

**Step 2**: After receiving  $\{MS5\}SK_{HP}$  and decrypting the message from 'P', 'H' generates  $SeK_{HS}$  from the  $SK_{HS}$  and  $RN_S$  and decrypts the received message  $\{MS3\} SeK_{HS}$ . It uses the assumptions from AS1 to AS8

$P \rightarrow H : \{MS5\}SK_{HP}$   
 $MS5 : \{ \{MS3\} SeK_{HS}, RN_S, \{MS4\} \}$   
 $\{MS4\} = ID_P, ID_S, T_P, N_P, LOC_P$

2) ASSUMPTIONS

'Z' is a set of participants {S, P, H and D}. CA issues certificates for their respective public keys to all the entities involved and all the entities have their symmetric, public and private keys (AS1, AS2).

- a) **AS1.CA** believes  $(\forall Z \in S, P, H \text{ and } D \xrightarrow{K_X} X) CA$  believes all the participants has their own public keys.
- b) **AS2.**  $Z \in \{S, P, H \text{ and } D\}$  Z believes  $(\xrightarrow{K_{CA}} CA)$ . All the SRSTMIS framework entities possess the public key and X.509 and short lived certificate of CA.
- c) **AS3.** H believes  $\# N_S$ , if H sees nonce generated by 'S'  $N_S$  in a message, then H considers that it is not a replay message.
- d) **AS4.** H believes  $\# N_P$ , if H sees nonce generated by 'P'  $N_P$  in a message, then H considers that it is not a replay message.
- e) **AS5.** D believes  $\# N_H$ , if 'D' sees nonce generated by 'H'  $N_H$  in a message, then D considers that it is not a replay message.

$MS3 : \{ID_P, ID_S, T_S, N_S, LOC_S, PR, H(PR)\}$   
**H believes P said** :  $\{MS5\}SK_{HP} \text{---}$  (3)  
**H believes # N<sub>S</sub>** from **AS3** --- (4)  
**H believes # N<sub>P</sub>** from **AS4** --- (5)  
**H believes # T<sub>S</sub>** from **AS6** --- (6)  
**H believes # T<sub>P</sub>** from **AS7** --- (7)  
 $ID_P \text{ (fromMS3)?} = ID_P \text{ (fromMS4)} \text{---}$  (8)  
 $ID_S \text{ (fromMS3)?} = ID_S \text{ (fromMS4)} \text{---}$  (9)  
 $T_S \text{ (fromMS3)?} = T_P \text{ (fromMS4)} \text{---}$   
 $LOC_S \text{ (fromMS3)?} = LOC_P \text{ (fromMS4)} \text{---}$  (11)  
 So, from the statements (3) to (11)  
**H believes**  $\{MS5\}SK_{HP}$

**STEP3** :  $H \rightarrow D: \{ID_P, ID_H, T_H, N_H, LOC_P, PR\}SK_{HD}$   
 'D' decrypts the received  $\{ID_P, ID_H, T_H, N_H, LOC_P, PR\}SK_{HD}$   
 from 'H'. It uses the assumptions from AS1 to AS8.  
**D believes H said** :  $\{ID_P, ID_H, T_H, N_H, LOC_P, PR\}SK_{HD}$  (8)  
**D believes # T<sub>H</sub>** from **AS8** --- (12)  
**D believes # N<sub>H</sub>** from **AS5** --- (13)  
 So, from the statements (12) to (13),  
**D believes**  $\{ID_P, ID_H, T_H, , LOC_P, PR\}SK_{HD}$   
 s

## B. FORMAL VERIFICATION OF THE SRSTMIS PROTOCOL USING THE SCYTHYR TOOL

SRSTMIS protocol is written in Security Protocol Description Language (SPDL); SPDL is a language for the Scyther simulation tool [19] and [20]; it verifies the security of protocols. Scyther tool defines the roles of SRSTMIS framework and all the entities involved face different types of attacks such as authentication attack, integrity attack and confidentiality attack. This tool helps in verifying, falsifying, and analyzing the security properties of SRSTMIS protocol with a unique ability to verify multi-protocol attacks.

**Attack model:** SRSTMIS framework is implemented in SE, UICC and TPM and their credentials are generated and stored in these tamper resistant devices; in addition to these WBC is implemented in these tamper resistant devices along with MHAs ensuring the safety and security of the keys. So all the entities involved in the framework which includes 'P', 'D' and 'H' ensures end to end security.

## V. SECURITY ANALYSIS

This section presents the security analysis of SRSTMIS protocol. Table 5 shows the comparative analysis of SRSTMIS with the related work.

- 1) **Proposition 1:** *The proposed protocol healthcare protocol ensures confidentiality property*  
*Proof:* Encrypted medical data is exchanged in SRSTMIS framework thereby ensuring confidentiality property.

- 2) **Proposition 2:** *The proposed protocol healthcare protocol ensures Mutual Authentication property*  
*Proof:* MPKI is an integral part of the proposed framework 'SRSTMIS' implemented in all the entities involved in 'SRSTMIS' thereby ensuring mutual authentication property.
- 3) **Proposition 3:** *The proposed protocol healthcare protocol ensures integrity property*  
*Proof:* SRSTMIS framework ensures integrity of the messages as the messages are encrypted and these encrypted messages timestamps and nonce thereby ensuring timeliness, freshness, uniqueness and integrity properties
- 4) **Proposition 4:** *The proposed protocol ensures the security of keys*  
*Proof:* 'SRSTMIS' is implemented in SE, UICC and TPM and their credentials are generated and stored in these tamper resistant devices; in addition to these WBC is implemented in these tamper resistant devices along with MHAs ensuring the safety and security of the keys.
- 5) **Proposition 5:** *The proposed healthcare framework provides Resists Configuration Tampering*  
*Proof:* 'SRSTMIS' withstands this attack by overcoming reverse engineering attacks on MHAs, by imposing Self-Signing Restriction on healthcare applications, by code and control obfuscation and these applications are digitally signed by the CA thereby resisting configuration tampering.
- 6) **Proposition 6:** *The proposed 'SRSTMIS' framework in the realm of healthcare is in compliance with HIPAA regulations*  
*Proof:* Our proposed SRSTMIS framework ensures the secrecy of keys, patient's medical data during rest and during the transit thereby ensuring HIPAA regulations.
- 7) **Proposition 7:** *The proposed 'SRSTMIS' framework in the realm of healthcare ensures anonymity of the patient from Doctor*  
*Proof:* In our proposed 'SRSTMIS' framework 'CA' and 'H' issues anonymous identity to 'P' in the form of 'TAC', so 'SRSTMIS' framework ensures anonymity of the 'P' from 'D'.
- 8) **Proposition 8:** *Proposed 'SRSTMIS' framework consumes fewer resources from the patient's perspective.*  
*Proof:* 'SRSTMIS' framework uses only shared symmetric key (AES) which consumes very less resources. We compared our proposed work with the related work and found to be consuming fewer resources than the existing works as shown in the section VII.
- 9) **Proposition 9:** *Proposed 'SRSTMIS' framework ensures Application and Communication security*  
*Proof:* MPKI and WBC are an integral part of the proposed 'SRSTMIS' framework. Application security is ensured using MPKI and WBC, communication

- security is ensured using SSL/TLS protocol in our proposed 'SRSTMIS' framework.
- 10) **Proposition 10:** *Proposed 'SRSTMIS' healthcare protocol is Formally verified*  
*Proof:* Proposed 'SRSTMIS' healthcare protocol is formally verified by BAN logic formal language and by Scyther formal tool our protocol overcomes all the known attacks.
  - 11) **Proposition 11:** *Proposed 'SRSTMIS' healthcare framework ensures Resistance Against Unauthorized Key Computation*  
*Proof:* Intruder/Attacker will not be able to compute/retrieve the symmetric keys as the patient/doctor/hospital as they cannot tamper the SE, UICC and TPM as the keys in these hardware devices implements WBC.
  - 12) **Proposition 12:** *Proposed 'SRSTMIS' healthcare framework provides Resistance to Multi-Protocol Attack*  
*Proof:* 'SRSTMIS' protocol was successfully verified using Scyther tool which proves that 'SRSTMIS' protocol resists Multi-Protocol Attack.
  - 13) **Proposition 13:** *Proposed 'SRSTMIS' healthcare framework provides Resistance to "Man-in-The-Middle" Attack*  
*Proof:* Intruder/Attacker will not be able to compute/retrieve the symmetric keys as the patient/doctor/hospital as they cannot tamper the SE, UICC and TPM as the keys in these hardware devices implements WBC, in addition to these exchanged messages contains timestamps and nonce which resists Man In The Middle Attacks.
  - 14) **Proposition 14:** *Proposed 'SRSTMIS' healthcare framework ensures Resistance to Replay Attack*  
*Proof:* Intruder/Attacker will not be able to compute/retrieve the symmetric keys as the patient/doctor/hospital as they cannot tamper the SE, UICC and TPM as the keys in these hardware devices implements WBC, in addition to these exchanged messages contains timestamps and nonce which resists replay Attacks.
  - 15) **Proposition 15:** *Proposed 'SRSTMIS' healthcare framework provides Resistance to Impersonation Attack*  
*Proof:* SRSTMIS framework withstands impersonation attacks as the intruder will not be able to retrieve/generate keys from the MHA/ SE/ UICC/TPM as these keys are protected by WBC.
  - 16) **Proposition 16:** *Proposed 'SRSTMIS' healthcare framework ensures Resistance to Parallel Session Attack*  
*Proof:* Intruder/Attacker will not be successful in starting a new parallel session in SRSTMIS as 'P' establishes a secure channel using certificate pinning and TLS protocol with 'H'. So, SRSTMIS framework resists to parallel session attack.
  - 17) **Proposition 17:** *Proposed 'SRSTMIS' healthcare framework ensures Resistance to Physically Stolen Device Attack*  
*Proof:* If an attacker/intruder/adversary is in possession of 'S', patient's smartphone (P) and TPM of 'H' he will not be able to retrieve patient's health data and credentials as they are protected by password and moreover all the credentials in these devices are protected by WBC. So our proposed 'SRSTMIS' healthcare framework Resists Physically Stolen Device Attack
  - 18) **Proposition 18:** *Proposed 'SRSTMIS' healthcare framework ensures Resistance Against Stolen Verifier Attack*  
*Proof:* SRSTMIS framework resists Stolen Verifier Attack as the intruder will not be able to retrieve/generate keys from the MHA/ SE/ UICC/TPM as these keys are protected by WBC.
  - 19) **Proposition 19:** *Proposed 'SRSTMIS' healthcare framework overcomes DoS and DDoS attacks*  
*Proof:* SRSTMIS framework resists DoS and DDoS attacks as 'H' detects these attacks by change-point detection, activity profiling and wavelet-based signal analysis detection techniques. In addition to these 'H' installs DoS/DDoS protection tools such as "Fort Guard Anti-DDoS".
  - 20) **Proposition 20:** *Proposed 'SRSTMIS' healthcare framework Resists Stolen Smart card attack*  
*Proof:* If an attacker is in possession of 'S', 'UICC' and 'TPM' the intruder will not be able to use these devices as Tampering will not possible. So 'SRSTMIS' healthcare framework resists stolen smart card attack.
  - 21) **Proposition 21:** *Proposed 'SRSTMIS' healthcare framework overcomes Outsider attacks*  
*Proof:* SRSTMIS framework resists Outsider Attacks as Intruder/Attacker will not be able to read/modify/tamper the messages as these are encrypted and the intruder retrieve the keys. In addition to these application and communication security are ensured in SRSTMIS framework.
  - 22) **Proposition 22:** *Proposed 'SRSTMIS' healthcare framework withstands Insider attacks*  
*Proof:* Assuming that a disgruntled staff/doctor tries to extract the shared symmetric keys from a patient's MHA, he will not succeed as the WBC ensures the security and safety of keys.
  - 23) **Proposition 23:** *Proposed 'SRSTMIS' healthcare framework withstands Repackaging attacks*  
*Proof:* 'SRSTMIS' healthcare framework overcomes repackaging attack on MHA by Code and control Obfuscation, Code Attestation and by imposing Self-Signing Restriction.
  - 24) **Proposition 24:** *Proposed 'SRSTMIS' healthcare framework withstands BlueBorne attack.*  
*Proof:* In Blue Borne attack, attackers exploit the Bluetooth vulnerabilities but 'SRSTMIS' healthcare framework withstands Blue Borne attack as encrypted

messages exchanged and WBC is implemented in MHA.

## VI. THREAT MODELLING

The process of threat modeling is divided into the three main phases as following: identifying assets, access points and trust levels, Identify and Rank all the potential threats and Discover countermeasures and build mitigation plan. Table 3 suggests a list of Threats and the countermeasures provided by our proposed framework corresponding to STRIDE.

### (1) Identifying assets, access points and trust levels:

An asset is a valuable thing which is owned by an entity of SRSTMIS framework, and the intruders/attackers/adversaries are interested in, and wish to retrieve/access, control or delete it. This step is the most crucial step in threat modeling. Access points are the interfaces through which intruders/attackers/adversaries can interact with the system in order to gain access to assets. The next step is to identify and define boundaries of trust in the system.

There are different levels of trust indicating the trust required for accessing a component from a system. The main idea of a trust boundary is that within a boundary, there is a common level of security, so within that boundary the components trust each other and will not question the integrity of each other.

**List of Assets in our proposed SRSTMIS framework:** Sensor, MHA in Sensor, Smart phone of the patient, MHA in Smart phone of the patient, TPM in hospital

**List of Access Points (AP) in our proposed SRSTMIS framework:** Smart phone of the patient,

**List of Trust Levels (TL) in our proposed SRSTMIS framework:** There are three trust boundaries in our proposed framework they are

- i) **User and Device boundary:** User and Device boundary is between Patient and the MHA in the UICC of the Patient's smartphone in which patient credentials are entered and the patient receives response from the MHA.
- ii) **Internet boundary:** Internet boundary is between Patient's Smartphone and the Hospital Server (H), Patient encrypts the messages using the shared symmetric key between 'P' and 'H' ensuring application security and the TLS is used in ensuring communication security.
- iii) **CorpNet Trust boundary:** CorpNet Trust boundary is between the Hospital Server (H) and the Hospital Database, messages are exchanged between them and protected using IPSec protocol. Hospital has Private Hospital Network (PHN) which hosts hospital and hospital database. PHN is a dedicated network which connects differed sub-entities in the hospital premises, outside traffic is not allowed in the PHN.

(2) **Identify and Rank all the potential threats:** The capabilities and objectives of an intruder which can arise

from inside or outside the organization are termed as threats. Threats can be identified by analyzing the assets and access points in the framework which compromise the security properties such as availability, mutual authentication, confidentiality, non-repudiation and integrity. Microsoft STRIDE model classifies threats into six classes.

- i) **Spoofing** is an attempt to gain access to a system by means of a false identity. Patient's smart phone of may be spoofed by an attacker/adversary which leads to data being written to the attacker/adversary's device instead of the patient's smart phone.
- ii) **Tampering** is a means of manipulation of data without the consent and permission of the data owner. Role-Based Access Control (RBAC) is deployed in our proposed healthcare framework., so tampering of data and logs are not possible. Use of SE, UICC and TPM. Tampering Data - patients and doctors intentionally/unintentionally can modify, update, and remove/delete patient's medical data.
- iii) **Repudiation** is the ability of authorized users denying of conducting specific actions.
- iv) **Unwanted exposure of confidential information** is called Information disclosure.
- v) The process of making a system or an application unavailable to its legitimate users is called Denial of service.
- vi) When a user with limited privileges elevates his/her privileges by identity theft in order to gain access to crucial assets of a system.

(3) **Discover countermeasures and build mitigation plan:** After identifying the assets and threats we need to have mechanisms and strategies in order to mitigate these threats. This phase provides a mitigation plan to overcome the identified threats.

- a) **Countermeasures for Spoofing:** In our proposed TMIS framework, Spoofing is not possible at Sensor (S), Smartphone (P) and at the Hospital (H) as all the entities involved in the framework have their credentials and MHA on SE, UICC and TPM which are tamper resistant. In addition to these all the entities adopt WBC which ensures the safety and security of keys and applications. Proposed TMIS framework ensures application and communication security.
- b) **Countermeasures for Tampering:** SRSTMIS framework ensures integrity of the messages as the messages are encrypted and these encrypted messages contain timestamps and nonce thereby ensuring timeliness, freshness, uniqueness and integrity properties. So tampering of messages is not possible. In our proposed TMIS framework Hospital Employs Logging and Monitoring Manager (LMM) which keeps track of the logging information. CA employs an auditor in the hospital who audits the working of the hospital.
- c) **Countermeasures for Repudiation:** In our proposed TMIS framework Hospital Employs Logging and

TABLE 2. Comparative analysis of our proposed work with related work.

Protocols Features	[7]	[8]	[9]	[13]	[2]	[3]	[4]	[10]	[11]	SRSTMIS (Our Proposed)
Confidentiality	x	x	x	x	x	x	x	x	x	✓
Mutual Authentication	x	x	x	x	x	x	x	x	x	✓
Integrity	x	x	x	x	x	x	x	x	x	✓
Provides security of keys	x	x	x	x	x	x	x	x	x	✓
Configuration Tampering	x	x	x	x	x	x	x	x	x	✓
Compliance with HIPAA Regulations	x	x	x	x	x	x	x	x	x	✓
Anonymity of patients	x	x	x	x	x	x	x	x	x	✓
Consumes fewer resources (Communication & Computational Cost)	x	x	x	x	x	x	x	x	x	✓
Ensures Communication Security	x	x	x	x	x	x	x	x	x	✓
Ensures Application Security	x	x	x	x	x	x	x	x	x	✓
Formal Verification	x	x	x	x	x	x	x	x	x	✓
Resists Unauthorized Key Computation	x	x	x	x	x	x	x	x	x	✓
Multi-Protocol Attacks	x	x	x	x	x	x	x	x	x	✓
Resists MITM Attack	x	x	x	x	x	x	x	x	x	✓
Resists Replay Attack	x	x	x	x	x	x	x	x	x	✓
Resists Impersonation Attack	x	x	x	x	x	x	x	x	x	✓
Resists Parallel Session Attack	x	x	x	x	x	x	x	x	x	✓
Resists Physically Stolen Device Attack	x	x	x	x	x	x	x	x	x	✓
Resists Stolen Verifier Attack	x	x	x	x	x	x	x	x	x	✓
Overcomes DOS & DDoS Attack	x	x	x	x	x	x	x	x	x	✓
Resists Stolen Smart Card Attack	x	x	x	x	x	x	x	x	x	✓
Outsider Attack	x	x	x	x	x	x	x	x	x	✓
Insider Attack	x	x	x	x	x	x	x	x	x	✓
Overcomes Repackaging Attacks	x	x	x	x	x	x	x	x	x	✓
BlueBorne Attack	x	x	x	x	x	x	x	x	x	✓

Monitoring Manager (LMM) which keeps track of the logging information. CA employs an auditor in the hospital who audits the working of the hospital. MHAs in the Sensor (S) and Smartphone (P) logs all the information sent and received.

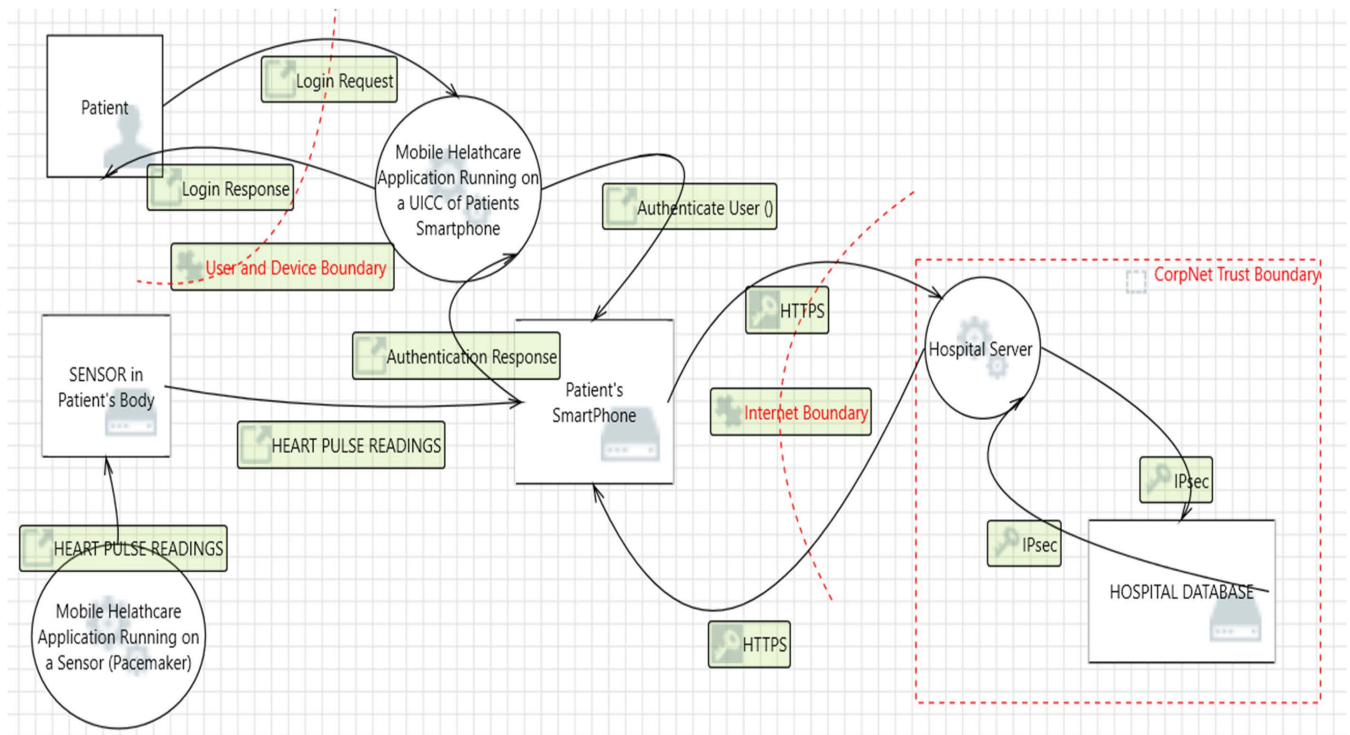
- d) **Countermeasures for Information Disclosure:** SRSTMIS framework adopts RBAC (Role Based Access Control). Encrypted medical data is exchanged in SRSTMIS framework thereby ensuring confidentiality property. SRSTMIS framework ensures integrity of the messages as the messages are encrypted and these encrypted messages contain timestamps and nonce thereby ensuring timeliness, freshness, uniqueness and integrity properties. So, SRSTMIS framework

ensures the secrecy and integrity of the patient’s data, so patient’s data cannot be compromised.

- e) **Countermeasures for Denial of service:**SRSTMIS framework resists DoS and DDoS attacks as ‘H’ detects these attacks by change-point detection, activity profiling and wavelet-based signal analysis detection techniques. In addition to these ‘H’ installs DoS/DDoS protection tools such as “Fort Guard Anti-DDoS”.
- f) **Countermeasures for Elevation of privilege:** SRSTMIS framework adopts RBAC (Role Based Access Control). Attacker will not be able to impersonate any of the entities involved in the ecosystem our proposed SRSTMIS framework ensures application and communication security. In addition to these

**TABLE 3. Threat modelling.**

STRIDE	Threats	Countermeasures Provided by SRSTMIS
<b>Spoofing</b>	<p>Spoofing of Sensor (S): Sensors (S) such as WHRMs may be spoofed by the attackers', leading to incorrect data being delivered to the hospital (H)</p> <p>Smartphone's (P) Spoofing: P's smartphone may be spoofed by the attackers which leads to patient's readings generated from 'S' will reach attackers smartphone instead of 'P'</p> <p>Spoofing of Hospital Servers (H): Hospital servers may be spoofed by the intruders, leading to incorrect data.</p>	<p>In our proposed TMIS framework, Spoofing is not possible at Sensor (S), Smartphone (P) and at the Hospital (H) as all the participants in SRSTMIS have their credentials and MHA on SE, UICC and TPM which are tamper resistant. In addition to these all the entities adopt WBC which ensures the safety and security of keys and applications.</p> <p>Proposed TMIS framework ensures application and communication security</p>
<b>Tampering</b>	<p>Patient's readings tampering – Any participant of SRSTMIS can intentionally/unintentionally modify, update, and remove/delete patient's medical data.</p> <p>Tampering of Log Files - patients, system administrators, doctors'/healthcare professionals can manipulate log files.</p>	<p>Role-Based Access Control (RBAC) is deployed in our proposed healthcare framework., so tampering of data and logs are not possible.</p> <p>Use of SE, UICC and TPM</p> <p>In SRSTMIS framework log file tampering is not possible it ensures communication and application security.</p>
<b>Repudiation</b>	Data repudiation – Any participant of the SRSTMIS framework denies sending, receiving and editing medical data.	In our proposed TMIS framework Hospital Employs Logging and Monitoring Manager (LMM) which keeps track of the logging information CA employs an auditor in the hospital who audits the working of the hospital MHAs in the Sensor (S) and Smartphone (P) logs all the information sent and received
<b>Information disclosure</b>	<p>Unauthorized disclosure of medical data.</p> <p>Lost/Stolen Sensor(S) and Smartphone (P) - patients losing their Sensors (S) and smartphones (P) can compromise their credentials</p>	<p>SRSTMIS framework adopts RBAC (Role Based Access Control)</p> <p>SRSTMIS framework overcomes stolen verifier attack as the attacker will not be successful in getting useful from the 'S', Smartphones or from hospital server as all the credentials are in tamper resistant hardware devices and protected by WBC.</p> <p>Use of SE, UICC and TPM</p> <p>SRSTMIS framework is free from tampering patient's readings as it ensures communication and application security.</p>
<b>Denial of Service (DoS) Attack</b>	It's an attack on the availability of Resources	<p>Our proposed SRSTMIS framework withstands DOS attacks as DoS attacks as 'H' installs DoS/DDoS protection tools such as "Fort Guard Anti-DDoS"</p> <p>Redundant components, SE, UICC and TPM</p> <p>Data rate limiting, Redundant data/network paths</p>
<b>Elevation of privilege</b>	Gaining Elevation of privileges by impersonating as doctor/ patient /administrator	<p>SRSTMIS framework adopts RBAC (Role Based Access Control)</p> <p>Attacker will not be able to impersonate any of the entities involved in the ecosystem our proposed SRSTMIS framework ensures application and communication security. In addition to these SRSTMIS is proposed on SE, UICC and TPM by adopting MPKI and WBC mechanisms.</p>



**FIGURE 6. Threat Modeling of our proposed framework using MTM Tool 2016.**

SRSTMIS is proposed on SE, UICC and TPM by adopting MPKI and WBC mechanisms.

### VII. IMPLEMENTATION AND PERFORMANCE ANALYSIS OF THE PROPOSED PROTOCOL

This section highlights the implementation details and performance analysis of the proposed protocol.

#### A. IMPLEMENTATION OF THE PROPOSED PROTOCOL

SRSTMIS is implemented in Android Studio using Kotlin language. AES symmetric encryption algorithm is used to ensure all the confidentiality property, Timestamps and Nonce ensures the freshness and timeliness of the messages transmitted. GCM mode is used in AES symmetric encryption algorithm which encrypts the patient's readings

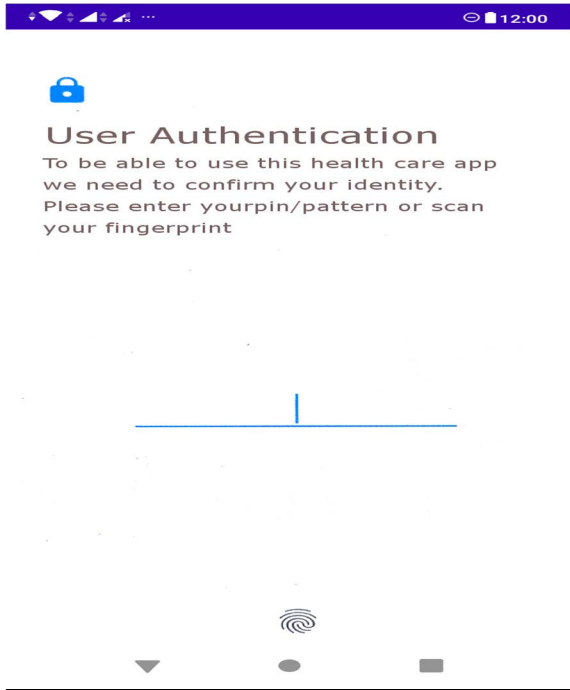


FIGURE 7. Patient authentication screen.

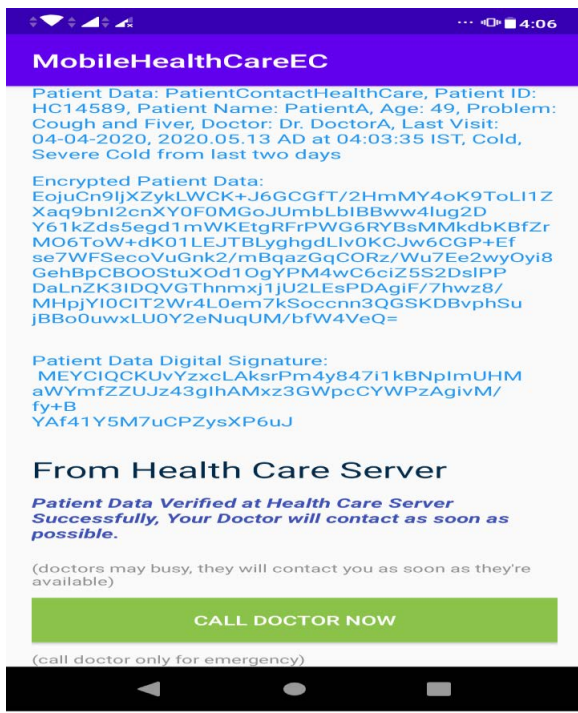


FIGURE 8. Confirmation message from hospital server.

1) PERFORMANCE ANALYSIS OF THE PROPOSED PROTOCOL

The efficacy of the SRSTMIS’s protocol is better than the existing TMIS solutions as it only employs symmetric encryption/decryption and one-way hash operations.

TABLE 4. Comparison of computational costs of our proposed protocol with related works.

Protocol	Cost for P’s smartphone	Cost for S	Overall computation cost	Overall execution time (sec)
[7]	4TH + 2TS	TH + 2TS	5TH + 4TS	0.5232
[8]	6TH + 2TS	5TH + 2TS	11TH + 4TS	0.5256
[9]	10TH+2TS	4TH+TS	14TH+2TS	0.2662
[13]	4TH+2TS	TH + 2TS	5TH + 4TS	0.5232
SRSTMIS	TS	TS	2TS	0.2606

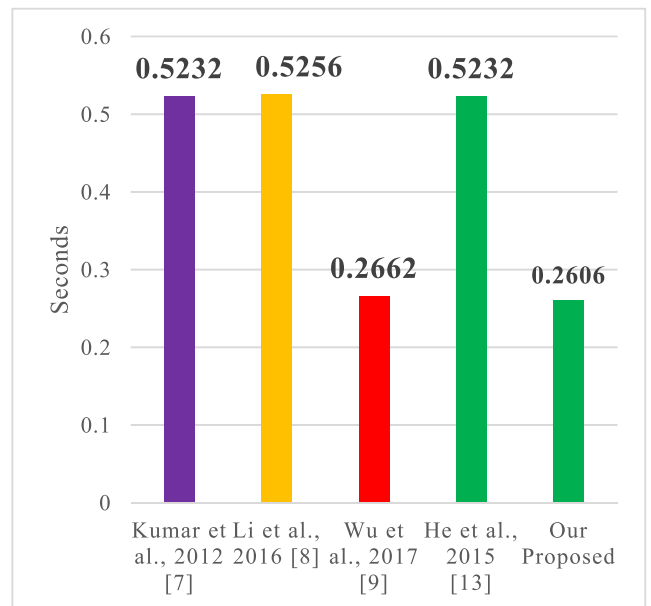


FIGURE 9. Bar chart for computational costs of the proposed protocol.

TABLE 5. Comparison of energy costs of our proposed protocol with related works.

Protocol	Cost for P’s smartphone	Cost for S	Overall energy cost	Overall Energy cost in $\mu$ Joules
[7]	4EH + 2ES	EH+ 2ES	5EH+ 4ES	8.64
[8]	6 EH + 2 ES	5 EH + 2 ES	11 EH + 4 ES	13.2
[9]	10 EH + 2 ES	4 EH + ES	14 EH + 3ES	14.27
[13]	4EH+2ES	EH+2ES	5EH + 4ES	8.64
SRSTMIS	ES	ES	2ES	2.42

According to [14] the time complexities are TH = 0.0004 and TS = 0.1303 where TH is time taken for hashing and TS is the time taken for encryption/decryption, TH and TS are



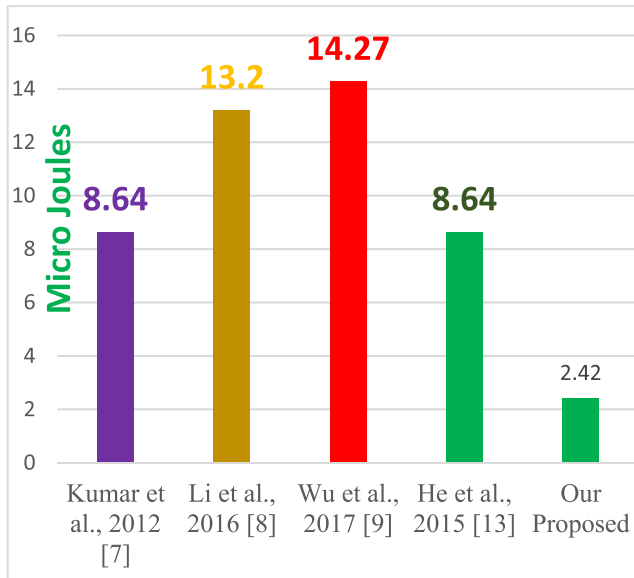


FIGURE 10. Bar Chart for energy costs of the proposed protocol.

in seconds. SRSTMIS's protocol has better performance as shown in figure 9, table 5 compares the energy consumption of SRSTMIS's protocol with the other existing works in the literature. In order to calculate energy consumption, we used hash and symmetric key operations, according to [15] the energy consumed in generating one encryption/decryption operation based on AES algorithm is 1.21 Micro Joules/byte and for generating one SHA-1 hash (EH) is 0.76 Micro Joules as shown in Figure 10.

## VIII. CONCLUSION

This paper proposes a TMIS architecture, a procedure to verify the safety and security of patients credentials and Mobile Healthcare Applications (MHA) and a secure protocol. We have threat modeled our proposed healthcare framework using STRIDE approach and successfully verified using Microsoft Threat Modeling tool 2016. Our proposed secure and lightweight authentication scheme has been successfully verified with BAN logic and Scyther tool, SRSTMIS withstands DDoS and DDoS attacks in addition to multi-protocol and Blue Borne, reverse engineering and Phlashing attacks. Proposed framework overcomes information leakage from sensors during rest and during transit, key loss from healthcare applications of the sensors and smart phone and false authentication among the entities involved in the system thereby ensuring HIPAA regulations. We have successfully implemented our protocol using kotlin language in Android Studio. SRSTMIS is better than the existing TMIS solutions. Safety and security of the keys are ensured by White-Box Cryptography (WBC). Proposed framework overcomes reverse engineering attacks. SRSTMIS's communication, computational and energy costs are far less than the existing TMIS solutions.

## ACKNOWLEDGMENT

The authors gratefully acknowledge the editor and the reviewers' helpful comments and suggestions, which have improved the presentation.

The authors deeply acknowledge the Researchers Supporting Program (TUMA-Project-2021-14), AlMaarefa University, Riyadh, Saudi Arabia, for supporting steps of this work.

Funding: Dr. Shaik Shakeel Ahamad would like to thank the Deanship of Scientific Research at Majmaah University for supporting this work under Project No. R-2022-295.

This research was supported by Researchers Supporting Program (TUMA-Project-2021-14), AlMaarefa University, Riyadh, Saudi Arabia.

Competing interests: The authors have declared that no competing interests exist.

## REFERENCES

- [1] E. Sazonov and M. R. Neuman, *Wearable Sensors: Fundamentals, Implementation and Applications*. Amsterdam, The Netherlands: Elsevier, 2014.
- [2] L. Xiao, S. Xie, D. Han, W. Liang, J. Guo, and W.-K. Chou, "A lightweight authentication scheme for telecare medical information system," *Connection Sci.*, vol. 33, no. 3, pp. 769–785, 2021, doi: [10.1080/09540091.2021.1889976](https://doi.org/10.1080/09540091.2021.1889976).
- [3] A. Tewari and B. B. Gupta, "An Internet-of-Things-based security scheme for healthcare environment for robust location privacy," *Int. J. Comput. Sci. Eng.*, vol. 21, no. 2, pp. 298–303, 2021, doi: [10.1504/IJCSE.2020.105742](https://doi.org/10.1504/IJCSE.2020.105742).
- [4] G. Sharma and S. Kalra, "A lightweight user authentication scheme for cloud-IoT based healthcare services," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 43, no. S1, pp. 619–636, Jul. 2019, doi: [10.1007/s40998-018-0146-5](https://doi.org/10.1007/s40998-018-0146-5).
- [5] M. Nicholls, *Cybersecurity Threat to Remote Monitoring Devices*. Accessed: Mar. 19, 2022. [Online]. Available: <https://healthcare-in-europe.com/en/news/cybersecurity-threat-to-remote-monitoring-devices.html>
- [6] P. K. Dhillon and S. Kalra, "Multi-factor user authentication scheme for IoT-based healthcare services," *J. Reliable Intell. Environments*, vol. 4, no. 3, pp. 141–160, Sep. 2018, doi: [10.1007/s40860-018-0062-5](https://doi.org/10.1007/s40860-018-0062-5).
- [7] P. Kumar, S. G. Lee, and H. J. Lee, "E-SAP: Efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks," *Sensors*, vol. 12, no. 2, pp. 1625–1647, 2012, doi: [10.3390/s120201625](https://doi.org/10.3390/s120201625).
- [8] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, and M. K. Khan, "A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity," *Secur. Commun. Netw.*, vol. 9, no. 15, pp. 2643–2655, Oct. 2016, doi: [10.1002/sec.1214](https://doi.org/10.1002/sec.1214).
- [9] F. Wu, L. Xu, S. Kumari, and X. Li, "An improved and anonymous two-factor authentication protocol for health-care applications with wireless medical sensor networks," *Multimedia Syst.*, vol. 23, no. 2, pp. 195–205, Mar. 2017.
- [10] F. M. Salem and R. Amin, "A privacy-preserving RFID authentication protocol based on El-gamal cryptosystem for secure TMIS," *Inf. Sci.*, vol. 527, pp. 382–393, Jul. 2020, doi: [10.1016/j.ins.2019.07.029](https://doi.org/10.1016/j.ins.2019.07.029).
- [11] H. Xu, J. Ding, P. Li, F. Zhu, and R. Wang, "A lightweight RFID mutual authentication protocol based on physical unclonable function," *Sensors*, vol. 18, no. 3, p. 760, Mar. 2018, doi: [10.3390/s18030760](https://doi.org/10.3390/s18030760).
- [12] A. Shaikh, M. S. Al Reshan, A. Sulaiman, H. Alshahrani, and Y. Asiri, "Secure telemedicine system design for COVID-19 patients treatment using service oriented architecture," *Sensors*, vol. 22, no. 3, p. 952, Jan. 2022, doi: [10.3390/s22030952](https://doi.org/10.3390/s22030952).
- [13] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, and S.-S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Syst.*, vol. 21, no. 1, pp. 49–60, Feb. 2015, doi: [10.1007/s00530-013-0346-9](https://doi.org/10.1007/s00530-013-0346-9).
- [14] L. Xu and F. Wu, "Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care," *J. Med. Syst.*, vol. 39, no. 2, Feb. 2015, doi: [10.1007/s10916-014-0179-x](https://doi.org/10.1007/s10916-014-0179-x).
- [15] S. L. Javan and A. G. Bafghi, "An anonymous mobile payment protocol based on SWPP," *Electron. Commerce Res.*, vol. 14, no. 4, pp. 635–660, Dec. 2014.

- [16] S. Muhammad, Z. Furqan, and R. K. Guha, "Understanding the intruder through attacks on cryptographic protocols," in *Proc. 44th Annu. Southeast Regional Conf. (ACM-SE)*, Melbourne, FL, USA, 2006, pp. 667–672.
- [17] M. Abadi, M. Burrows, C. Kaufman, and B. Lampson, "Authentication and delegation with smart-cards," *Sci. Comput. Program.*, vol. 21, no. 2, pp. 93–113, Oct. 1993.
- [18] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, Feb. 1990.
- [19] C. J. F. Cremers, "Scyther: Semantics and verification of security protocols," Ph.D. dissertation, Dept. Comput. Sci., Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2006, doi: [10.6100/IR614943](https://doi.org/10.6100/IR614943).
- [20] C. Cremers and P. Lafourcade, "Comparing state spaces in automatic security protocol verification," Ph.D. dissertation, Dept. Comput. Sci., Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2007.
- [21] M. Beunardeau, A. Connolly, R. Geraud, and D. Naccache, "White-box cryptography: Security in an insecure environment," *IEEE Secur. Privacy*, vol. 14, no. 5, pp. 88–92, Sep. 2016.
- [22] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14757–14767, 2017.
- [23] *Application Security Market Size Will Reach US 11 Billion by 2024—MarketWatch*. Accessed: Feb. 18, 2022. [Online]. Available: <https://www.marketwatch.com/press-release/application-security-market-size-will-reach-us-11-billion-by-2024-2019-05-06>
- [24] S. Singh. *IoT Medical Devices Market Worth \$63.43 Billion by 2023*. Accessed: Feb. 18, 2022. [Online]. Available: <https://www.marketsandmarkets.com/PressReleases/iot-medical-device.asp>
- [25] J. Herzog, "A computational interpretation of Dolev—Yao adversaries," *Theor. Comput. Sci.*, vol. 340, no. 1, 13, Jun. 2005, pp. 57–81.
- [26] *Traceable Anonymous Certificate*. document RFC 5636, Accessed: Feb. 8, 2022. [Online]. Available: <https://tools.ietf.org/html/rfc5636>
- [27] X. Li, F. Wu, M. K. Khan, L. Xu, J. Shen, and M. Jo, "A secure chaotic map-based remote authentication scheme for telecare medicine information systems," *Future Gener. Comp. Syst.*, vol. 84, pp. 149–159, Jul. 2018, doi: [10.1016/j.future.2017.08.029](https://doi.org/10.1016/j.future.2017.08.029).
- [28] A. K. Sutrala, A. K. Das, V. Odelu, M. Wazid, and S. Kumari, "Secure anonymity-preserving password-based user authentication and session key agreement scheme for telecare medicine information systems," *Comput. Methods Programs Biomed.*, vol. 135, pp. 167–185, Oct. 2016, doi: [10.1016/j.cmpb.2016.07.028](https://doi.org/10.1016/j.cmpb.2016.07.028).



**SHAIK SHAKEEL AHAMAD** received the dual Ph.D. degrees in computer science (cyber security) and realm of secure mobile payment protocols and formal verification from the University of Hyderabad and the Institute for Development and Research in Banking Technology (IDRBT), Hyderabad, India, respectively. He was a Professor at the Department of CSE and the Head of Computer Networks and Security Research Group, KL University, India. He is currently working as an

Assistant Professor at the Department of Information Technology, College of Computer and Information Sciences, Majmaah University, Al-Majmaah, Saudi Arabia. His research interests include cyber security, cloud computing, secure mobile payments, block chain technology, secure smart grids, and security and privacy in healthcare 4.0. He is a Certified EC Council Instructor (CEI), an EC Council Certified Security Analyst (ECSA), a Computer Hacking Forensic Investigator (CHFI), a Certified Ethical Hacker (CEH 11), a Certified Threat Intelligence Analyst (CTIA), and a Certified Application Security Engineer (CASE)—Java. He is serving as a review committee member for many ISI indexed journals.



**MOHAMMED AL-SHEHRI** received the B.S. degree from King Saud University, in 2001, the M.S. degree in computer and communication engineering from the Queensland University of Technology (QUT), Australia, in 2007, and the Ph.D. degree in information technology from Griffith University, Australia, in 2013. He has been an Associate Professor and the Dean at the College of Computer and Information Sciences, Majmaah University, Saudi Arabia, since 2015.

He was a Consultant of the Ministry of Defense, from 2013 to 2015. From 2002 to 2009, he worked at the Ministry of Defense, Saudi Arabia, eventually as an IT Manager. His research interests include computer science and information technology. He is also working in the field of education robotics, cloud computing, the IoT, artificial intelligence, data science, computer networks, security, and cybersecurity.



**ISMAIL KESHTA** received the B.Sc. and M.Sc. degrees in computer engineering and the Ph.D. degree in computer science and engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 2009, 2011, and 2016, respectively. He was a Lecturer at the Computer Engineering Department, KFUPM, from 2012 to 2016. Prior to that, he was a Lecturer at Princess Nourah Bint Abdulrahman University and Imam Muhammad Ibn Saud Islamic University, Riyadh, Saudi Arabia, in 2011. He is currently an Assistant Professor at the Computer Science and Information Systems Department, AlMaarefa University, Riyadh. His research interests include software process improvement, modeling, and intelligent systems.

...