REAS-TMIS: Resource-Efficient Authentication Scheme for Telecare Medical Information System

MUHAMMAD TANVEER1, ABD ULLAH KHAN2, (Member, IEEE), AHMED ALKHAYYAT3, (Member, IEEE), SHEHZAD ASHRAF CHAUDHRY4, YOUSAF BIN ZIKRIA5, (Senior Member, IEEE), AND SUNG WON KIM5

1School of Systems and Technology (SST), University of Management and Technology, Lahore 54700, Pakistan
2Department of Computer Science, National University of Sciences and Technology Balochistan Campus, Quetta 87300, Pakistan
3Department of Computer Technical Engineering, College of Technical Engineering, Islamic University, Najaf 54001, Iraq
4Department of Computer Engineering, Istanbul Gelisim University, 34310 Istanbul, Turkey
5Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, South Korea

Corresponding author: Sung Won Kim (swon@yu.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2021R1A6A1A03039493.

ABSTRACT The phenomenal growth of smartphones and wearable devices has begun crowd-sourcing applications for the Internet of Things (IoT). E-healthcare is considered the essential service for crowd-sourcing IoT applications that help remote access or storage medical server (MS) data to the authorized doctors, patients, nurses, etc., via the public Internet. As the public Internet is exposed to various security attacks, remote user authenticated key exchange (AKE) has become a pressing need for the secure and reliable use of these services. This paper proposes a new resource-efficient AKE scheme for telecare medical information systems, called REAS-TMIS. It uses authenticated encryption with associative data (AEAD) and a hash function. AEAD schemes are devised specifically for encrypted communication among resource-constricted IoT devices. These features of AEAD make REAS-TMIS resource-efficient. Moreover, REAS-TMIS dispenses with the elliptic curve point multiplication and chaotic map that are computationally expensive operations. In addition, REAS-TMIS renders the functionality of session key (SK) establishment for future encrypted communication between MS and users after validating the authenticity of the user. The security of SK is corroborated employing the well establish random oracle model. Moreover, Scyther-based security corroboration is implemented to show that REAS-TMIS is secure, and informal security analysis is executed to show the resiliency of REAS-TMIS against various security attacks. Besides, a thorough analysis shows that REAS-TMIS, while accomplishing the authentication phase, requires less computational, communication, and storage resources than the related authentication protocol.

INDEX TERMS Security, AEAD, e-healthcare, privacy, authentication, TIMS, smart city.

I. INTRODUCTION

The Internet of Things (IoT) evolution has impacted the essence of human life in different directions by providing significant acumen’s, productivity, and cost-effectiveness [1], [2]. Consequently, many novel applications essential for smart city environment and Industry 4.0 have been created. For instance, healthcare sector incorporates IoT to advance patient monitoring with reduced cost and thereby strengthens innovation in patients’ care. Essentially, the synthesis of IoT in the production and consumer sector is attributed to Industry 4.0. Similarly, Medicine 4.0 and Healthcare 4.0, the two major revolutions created by IoT for smart city environment, are boomed in healthcare sector, that has empowered innovative solutions for monitoring remote patient, dispensing medications, designing early warning and dynamic treatment strategies, and managing and maintaining medical equipment [3].

As one of the crucial applications of IoT in smart city environment, e-healthcare system is increasingly being used by the people all around the globe. Under certain circumstances, sharing the information associated with a patient with a group of medical professionals is essential to improving the treatment procedures [4]. For treatments where many
specialists are concerned, crowd-sourcing the IoT in e-healthcare services is needed. Fig. 1 presents a design of IoT applications where the crowd-sourcing IoT for e-healthcare is necessitated. In this design, the gateway node acts as the interface between the medical server (MS) and the remote users. MS is the main component of e-healthcare system from where diverse users like doctors, nurses, patients, medical policymakers, legal authorities, and insurance agents retrieve and deposit medical information. The contemporary conception of smart mobile devices (MDs) has yielded crowd-sourcing IoT applications. Data collected by MDs can be further processed to assist intelligently in different promising services. In e-healthcare applications, data accumulated by MDs are saved in different MSs. An authorized user accesses the information stored on MSs for monitoring and diagnosing purposes via the public Internet. The information when being accessed by the user through the public Internet is prone to be attacked by pernicious users and intruders. Hence, a resource-efficient and reliable security scheme for crowd-sourcing in e-healthcare services require consideration to preserve the vital and private medical information associated with the patient. This requires designing remote users’ authenticated key exchange (AKE) schemes to render secure access of sensitive resources to valid users [3]–[5].

A. RELATED WORK

Various AKE schemes have been proposed in the existing literature to enable secure and privacy-preserving communication within telecare medical information systems (TMIS). An AKE scheme checks the authenticity of the user and establishes a session key (SK) to enable encrypted communication between the medical server (MS) and the user. For this purpose, Kumari et al. [6] devised an elliptic curve cryptography (ECC)-based AKE scheme to enable a user to access the information from MS securely. However, the scheme cannot prevent password guessing (PGU), smart card/device loss (SMCL), user anonymity (URA), privilege insider (PIN), user impersonation (URIM), and de-synchronization (D-SYN) attacks. Khatoon et al. [7] proposed a user bi-linear-pairing (BP) based AKE scheme for TMIS. However, their scheme is incapable of thwarting URIM and PIN attacks and cannot provide URA feature. Similarly, the AKE scheme presented by Li et al. [8] is unable to impede PGU, IDGU, URIM, PIN, and SMCL attacks. Das et al. [9] proposed an SHA-based scheme, which cannot thwart server impersonation (SIM), man-in-the-middle (MATM), URIM, and PIN attacks and is unable to provide URA property.

The user AKE scheme proposed by Madhusudhan et al. [10] cannot resist replay, MATM, PIN, and SIM, and does not provide Mutual authentication (MA) and URA features. The AKE scheme presented in [11] is incapable of resisting denial-of-service (DoS), PIN, and masquerade attacks and does not provide URA and MA features. The authors proposed an AKE scheme in [12], which is prone to ephemeral secret leakage (EPLE), DoS, and key compromised attacks. The scheme presented by Garg et al. [13] in 2019, was proved insecure against key compromise impersonation and it was also argued in [14] that Garg et al’s scheme does not provide meter anonymity and forward secrecy. Similarly, the authors in [15], [16] presented the AKE schemes using an authenticated encryption with associative data (AEAD) and secure hash algorithm (SHA). However, their schemes cannot encompass all the security requirements stipulated by resource constrained IoT devices deployed for TMIS. A detailed summary of the various user AKE protocol for the TMIS environment is given in Table 1.

B. MOTIVATION

As described in Table 1, most of the schemes proposed to ensure indecipherable communications in the TMIS are unprotected against SIM, URIM, EPLE, and DoS attacks. In addition to this, some of the schemes are incapable of thwarting the D-SYN, PIN, and do not render the features, such as URA and MA. It is worth noting that public key cryptography and chaotic map-based user AKE scheme require significantly high computational resources because modular exponentiation and elliptic curve cryptography (ECC) based point multiplication operations are computationally expensive for the resource limited IoT devices. However, symmetric-key cryptography [41] is a feasible option for such devices. Stating more precisely, the recently proposed authenticated encryption with associative data cryptographic primitive are specifically designed for the resource constrained IoT devices. An AEAD scheme is efficient in terms of computational resource requirements and is therefore designed explicitly for resource-limited devices. In addition, an AEAD scheme provides the confidentiality, authenticity, and integrity of the data simultaneously. Therefore, using an AEAD scheme can reduce the computational time required to complete the authentication phase by reducing the cryptographic operation involved in the authentication process. Therefore, by leveraging the benefits of an AEAD scheme and hash function, we propose a lightweight and secure AKE scheme for the TIMS with the following contributions [42], [43].

C. RESEARCH CONTRIBUTION

1) We propose a resource-efficient authentication scheme for the TMIS, called REAS-TMIS, that utilizes the lightweight cryptography-based authenticated encryption with associative data (ASCON) and hash function “Esch256”. REAS-TMIS enables users and servers to set up SK for indecipherable communication after accomplishing the mutual authentication to ensure encrypted communication between users and medical servers. Moreover, REAS-TMIS ensures the anonymity and privacy of the user during the accomplishment of the AKE phase.

2) We leverage the Random oracle model (ROM) to validate the authenticity of the established SK. In addition, we utilize Scyther-based analysis and illustrate that REAS-TMIS is secure and resilient against various
covert security threats, including MATM, replay, D-SYN, URIM, SIM, and SMCL attacks.

3) We show that, in addition to rendering comparatively enhanced security functionalities, REAS-TMIS accomplishes the AKE process with the requirement of 54.04% lower computational and 19.79% lesser communication costs than the related AKE scheme.

The remaining of this paper is organized as follows. System models are elaborated in Section II. The proposed REAS-TMIS is explained in Section IV. The security validation is presented in Section V. The efficiency and effectiveness of REAS-TMIS are described in Section VI. Finally, the paper ends with concluding remarks in Section VII.

II. SYSTEM MODEL

A. NETWORK MODEL

The network model presented in Fig. 1 is considered for the proposed REAS-TMIS. The model comprises registration center (RC), medical server (MS), and users ($UR_x$ | $x = 1, 2, 3, \ldots, K$), where $K$ is the number of users. The users can be doctors, nurses, or family members authorized to access the information stored at MS. RC is responsible for the deployment of MS. Moreover, RC is also responsible for the registration of $UR_x$ before giving them access to the network resources, i.e., to view the patient record and the availability of other services provided by the medical center. MS stores all the information related to the health of a patient, which
are obtained from the patient monitoring system. In addition, MS stores the sensitive registration information associated with the URs. It is often the case that URs requires the data/information stored at MS. Thus, a security mechanism is required to enable safe communication between URs and MS. To provide URs with secure access to the system resources, an AKE scheme is required.

B. ATTACK MODEL

The Dolev–Yao (DY) model is considered as the threat model for the proposed REAS-TMIS. Under the DY model, an adversary, denoted by \( \mathcal{A} \), has the capabilities of seizing all the messages exchanged during the AKE phase. In addition to this, \( \mathcal{A} \) can capture the message or drop it, update the message content, and can re-transmit the modified message. Moreover, the smart user device is not considered to be a trusted device because \( \mathcal{A} \) can capture the user’s smart device and can procure the sensitive information stored in the memory of the device or smart card. Similarly, MS is considered to be placed in a secure environment, and \( \mathcal{A} \) can not capture it physically. However, the insider \( \mathcal{A} \) can retrieve the sensitive information stored in the database of MS and can perform various malicious activities on behalf of a particular user. Furthermore, we employ the postulates of the CK-adversary model. It is often the case that the postulates of the CK-adversary model hold simultaneously. The encryption and decryption processes of the ASCON can be expressed as follows.

\[
(CT, \text{Tag}) = E_{\text{Ke}}[(IV, AD, PT)] \\
(PT, \text{Tag}') = D_{\text{Ke}}[(IV, AD, CT)]
\]

where \( CT, \text{Tag}/Tag', Ke, IV, AD, \) and \( PT \) denote the ciphertext, authentication Tag, key, initialization vector, associative data, and plaintext, respectively.

C. Esch256

Esch256 is a lightweight hash algorithm that is designed for resource-constricted IoT devices. Moreover, Esch256 provides high security than SHA-160 with reduced computational cost. We denote the Esch256 hash operation by the expression \( H(\cdot) \). The detail description of “Esch256” hash function can be found in [49].

IV. THE PROPOSED REAS-TMIS SCHEME

The details of the proposed REAS-TMIS are presented in this section. REAS-TMIS comprises four phases: UR registration phase, AKE phase, password update (PUD) phase, and revocation (RV) phase. Table 2 tabulates a list of notations used to elaborate REAS-TMIS. The following subsections present the working of REAS-TMIS.

A. INITIALIZATION PHASE

RC is the trusted authority and is responsible for registering URs and MS. Before the deployment of MS in the target field, RC picks a unique identity \( ID_{MS} \) and a secret master key for \( MK_{MS} \) for MS. In addition, RC loads the credentials \( \{ID_{MS}, MK_{MS}\} \) in the temper resistance database of MS.

B. UR REGISTRATION PHASE

In user registration (URR) phase, URs needs to register with RC. RC assigns secret credentials to URs during the URR phase. Before accessing the network resources, URs needs to authenticate itself with MS. RC accomplishes the following imperative steps to register URs.

1) STEP URR-1: URs picks its own identity \( ID_{URs} \), random number \( R_{URs} \), and password \( PSW_{URs} \). In addition, URs by employing FE computes \( BIK_{URs} \cdot RP = Gen(BIO_{URs}) \), where the size of \( BIK_{URs} \) is 128 bits. Moreover, URs calculates \( W_1 = H(ID_{URs} || PSW_{URs}) \) and \( P_1 = (W_1^t + W_2^t) \) sends \( P_1 \) to RC.

Remark 1: Most of the AEAD schemes takes AD, nonce and key of sizes 128 bits. Here \( P_1 \) is obtained by performing
Exclusive-OR on $W^a_2$ and $W^b_2$, which are two chunks of $W_1$. Now, the size of $P_1$ has become 128 bits where as the size of $W_1$ was 256 bits. To make all the parameters compatible with AEAD encryption scheme (ASCON), we will perform the above operation.

2) STEP URR-2
RC on procuring $P_1$, picks random number $NP$, and computes $MP_1 = H(ID_{MS} \lor MK_{MS})$. $TID_{UR}, = (P_1 \lor NP)$, $PID = (TID_{UR}, \lor NP) \oplus MP_1$, $W_2 = H(P_1 \lor MK_{MS})$, and searching identity $SID = (W^a_2 \oplus W^b_2)$. Moreover, RC computes $W_3 = H(MK_{MS} \lor P_1 \lor ID_{MS})$ and secret parameter $SP_{UR} = (W^a_3 \oplus W^b_3)$ and sends $\{PID, SP_{UR}\}$ to $UR_s$. In addition, RC, stores the credentials $\{SID, SP_{UR}\}$ in MS’s database.

3) STEP URR-3
On procuring the secret credentials form RC, $UR_s$ computes $AD_1 = R_{UR}$, and $Ke_1 = P_1 \oplus BIK_{UR}$. Moreover, $UR_s$ by using ASCON encryption computes $\{CT_{UR}, Tag_{UR}\} = E_{Ke_1}[(AD_1), PT_{UR}]$, where $PT_{UR} = \{PID, SP_{UR}\}$. Finally, $UR_s$ stores the parameters $\{CT_{UR}, Tag_{UR}, Gen(\cdot), Rep(\cdot), RP, R_{UR}\}$ in its own memory.

C. AKE PHASE
In this phase, $UR_s$ performs the local authentication by validating its secret credentials and then sends the AKE request to MS. After achieving the mutual authentication both $UR_s$ and MS establish SK to achieve the indecipherable communication. Following steps are imperative to execute to accomplish the AKE process.

1) STEP AKE-1
inputs password $PSW^l_{UR_s}$, and identity $ID^l_{UR_s}$, imprints biometric $BIO_{UR_s}$, and computes the followings

$$BIK^l_{UR_s} = Rep(BIO^l_{UR_s}, RP),$$

$$P_1 = H(ID^l_{UR_s} \lor PSW^l_{UR_s}) and AD_1 = R_{UR_s},$$

$$Ke_1 = (P^l_1 \oplus P^b_1) \oplus BIK^l_{UR_s},$$

$$\{PT_{UR}, Tag_1\} = D_{Ke_1}(\{AD_1\}, CT_{UR}),$$

where $BIK^l_{UR_s}$ is the bio-metric key of associated with $UR_s$, which is obtained by using $Rep(\cdot)$ function of of $FE$. The parameter $P_1$ is determined by performing hash operation on $ID_{UR}$, and $PSW_{UR_s}$. Moreover, the secret encryption key $Ke_1$ is determined by concatenating $(P^l_1 \oplus P^b_1)$ and $BIK^l_{UR_s}$, where $P^l_1$ and $P^b_1$ are derived from $P_1$. Furthermore, $PT_{UR}, and \ Tag_1$, are the output of the ASCON decryption algorithm. Finally, smart user device $UD_x$ checks the the following condition

$$Tag_{1} \neq Tag_{UR_s}.$$  (7)

If the condition does not hold, $UD_x$ promptly terminates the AKE process and generates the login failure message. Otherwise, $UD_x$ retrieves $PT_{UR} = \{PID, SP_{URs}\}$ and proceeds with the AKE process and picks $R_a, R_b$, and $TS_1$. In addition to this, $UD_x$ computes

$$P_3 = H(PID \lor R_b \lor TS_1),$$

$$AD_2 = (P^a_3 \oplus P^b_3),$$

$$Ke_1 = (P^a_1 \oplus P^b_1),$$

$$\{CT_1, Tag_{1}\} = E_{Ke_1}[(AD_2), R_a],$$

where $Ke_1$ is secret key used in the encryption process. Finally, $UD_x$ constructs the message $MES_1 : [TS_1, PID, CT_1, Tag_1, R_b]$ and disseminates $MES_1$ to MS through open channel.

2) STEP AKE-2
On receiving $MES_1$, MS corroborates the freshness of the $MES_1$ by checking the condition $T_d \geq |Tre - TS_1|$, where $T_d$ is the allowed message delay, $Tre$ represents the $MES_1$’s receive time, and $TS_1$ signifies $MES_1$’s generation time. If it holds, MS computes

$$MP_2 = H(ID_{MS} \lor MK_{MS}),$$

$$MP_2 \oplus PID = (TID_{UR}, \lor NP),$$

$$P_1 = (TID_{UR}, \lor NP),$$

$$Ke_2 = (P^a_1 \oplus P^b_1),$$

$$P_3 = H(PID \lor R_b \lor TS_1),$$

$$AD_3 = (P^a_3 \oplus P^b_3),$$

$$(R_a, Tag_2) = D_{Ke_2}[(AD_3), CT_1],$$

where $MP_2$ is parameter generated by using the hash function with inputs $ID_{MS}$ and $MK_{MS}$. The parameter $P_1$ is obtained from (14) and $Ke_2$ is derived in (15), where $P_1$ is divided into two chunks. The parameter $P_3$ is obtained from (16) and $AD_3$ is determined from (17), where $P_3$ is divided in to two chunks. Finally, by using ASCON decryption process, MS generates the parameter $Tag_2$. In addition, MS validates $Tag_2 \neq Tag_1$, if holds, MS retrieves $R_a$. Moreover, MS computes

$$Q_5 = H(P_1 \lor MK_{MS}) and SID = (Q^a_5 \oplus Q^b_5).$$  (19)

The parameter $Q_5$ is obtained by performing the hash operation on the parameter $P_1$ and $MK_{MS}$ and $SID$ is derived after performing XORing $Q^a_5$ and $Q^b_5$, which are two parts of $Q_5$. $SID$ is used to retrieves the secret parameter $SP_{UR_s}$ from the database of MS. Furthermore, MS picks $TS_2, R_c$, and $NP_2$, and computes

$$QM = H(MK_{MS} \lor P_1 \lor ID_{MS}),$$

$$Ke_3 = (QM^a_1 \lor QM^b_1) \oplus R_a,$$

$$TID^a_{UR_s} = (P_1 \lor NP_2),$$

$$PID^{new} = (TID^b_{UR_s} \lor NP_2) \lor MP_2,$$

$$SK_{MS} = H(P_3 \lor P_1 \lor R_a \lor R_c \lor PID^{new}),$$

$$SK_v = (SK^a_{MS} \lor SK^b_{MS}),$$

$$PT_{MS} = (PID^{new} \lor R_c \lor SK_v),$$

$$(CT_2, Tag_3) = E_{Ke_3}[(AD_4), PT_{MS}],$$

VOLUME 10, 2022
where $K_{e3}$ is the secret key which is used in the encryption process, which is derived by splitting $QM$ into two parts. $PID_{new}^{new}$ is the new pseudo identity, which is will be used $UR_{x}$ to accomplish the new AKE session. In addition, we derive as $AD_{3} = AD_{4}$, where $AD_{3}$ is derived in (17). $SK_{SM}$ denotes a session key, which is used to encrypt the communication with $UR_{x}$. $SK_{v}$ is the session key verification parameter and its size is 128 bits, which used to validate the SK at user side. The parameter $PT_{MS}$ denotes the plaintext, which is generated by concatenating $R_{x}, PID_{new}^{new}$, and $SK_{v}$. Moreover, $MS$ by using ASCON encryption algorithm generates the parameters $CT_{2}$ and $Tag_{3}$. Finally, $MS$ contrives the message $MES_{2} \{TS_{2}, CT_{2}, Tag_{3}\}$ and dispatches $MES_{2}$ to $UR_{x}$ via open communication channel.

3) STEP AKE-3

On receiving $MES_{2}, UD_{x}$ checks the freshness of $MES_{2}$ by validating the condition $TD_{x} \geq |\text{Tre} - TS_{2}|$. If $MES_{2}$ is fresh, $UD_{x}$ determines $AD_{5} = AD_{1}$ and $Ke_{4} = (SP_{UR_{x}} \oplus R_{o})$. In addition, By using ASCON decryption algorithm, $UD_{x}$ computes

$$(PT_{MS}, Tag_{4}) = D_{Ke_{4}}(\{AD_{5}, CT_{2}\})$$

and checks the condition $Tag_{4} = Tag_{3}$, if it holds, $UD_{x}$ considers $MES_{2}$ as a valid message and procures $PT_{MS} = (PID_{new}^{new} \parallel K_{C} \parallel SK_{v})$, which is the plaintext. Moreover to ensure the encrypted communication with $MS$, $UD_{x}$ computes SK as follows

$SK_{UR_{x}} = H(P_{3} \parallel P_{1} \parallel R_{o} \parallel R_{e} \parallel PID_{new}^{new})$. (29)

Furthermore, $UD_{x}$ computes $SK_{v1} = (SK_{UR_{x}}^{a} \oplus SK_{UR_{x}}^{b})$ and checks the condition $SK_{v} = SK_{v1}$. If the condition is satisfied, $UD_{x}$ updates $PID$ with $PID_{new}^{new}$ and determines $PT_{UR_{x}}^{new} = \{PID_{new}^{new}, SP_{UR_{x}}\}$. In addition to this, $UD_{x}$ picks $R_{new}^{o}$ and computes

$$(CT_{new}^{o}, Tag_{UR_{x}}^{o}) = E_{Ke_{x}}(\{AD_{6}, PT_{new}^{o}\})$$

where $AD_{6} = R_{new}^{o}$. Finally, $UD_{x}$ updates $\{R_{UR_{x}}, CT_{UR_{x}}, Tag_{UR_{x}}\}$ with $\{R_{new}^{o}, CT_{new}^{o}, Tag_{new}^{o}\}$ in its own memory. The AKE phase of REAS-TMIS is summarized in Fig.2.

D. RV PHASE

If an adversary loses his smart device or card, $UR_{x}$ can procure new device as follows. To accomplish RV phase, $UR_{x}$ needs to compute $P_{1} = H(ID_{UR_{x}} \parallel PSW_{UR_{x}}^{o})$ and sends $P_{1}$ to RC. RC derives $SID$ as $SID = (W_{2}^{o} \oplus W_{5}^{o})$. In addition to this, RC searches $SID$ from the database of MS, if it is found, MS removes the record related to $SID$. After that $UR_{x}$ start the new registration process. For the new registration process we follow the same process as executed in Step URR-1 to Step URR-3.

E. PUD PHASE

To enhance the security of TMIS, it is necessary for $UR_{x}$ to update its password frequently. The proposed REAS-TMIS renders the functionality. $UR_{x}$ need to execute the following necessary step to update its password.

1) STEP PUD-1

$UR_{x}$ enters its old secret credentials, such as $PSW_{UR_{x}}^{o}$ and $ID_{UR_{x}}$, and imprints its bio-ometric information $BIO_{UR_{x}}^{o}$ at the available interface of $UD_{x}$. Moreover, $UD_{x}$ computes $BIO_{UR_{x}}^{o} = R_{o}(BIO_{UR_{x}}^{o}, RP_{o})$, $P_{o} = H(ID_{UR_{x}} \parallel PSW_{UR_{x}}^{o})$, $AD_{6} = R_{UR_{x}}^{o}Ke_{o} = (P_{o}^{o} \oplus P_{o}) \oplus BIO_{UR_{x}}^{o}$, $(PT_{UR_{x}}, Tag_{UR_{x}}^{o}) = D_{Ke_{o}}(\{AD_{6}, CT_{UR_{x}}\})$. Finally, $UD_{x}$ validates condition $Tag_{UR_{x}}^{o} = Tag_{UR_{x}}$. If it holds, $UD_{x}$ sends a message to $UR_{x}$ to enter new secret credentials.

2) STEP PUD-2

$UR_{x}$ picks new random number $R_{UR_{x}}^{n}$ and password $PSW_{UR_{x}}^{n}$. In addition, $UR_{x}$ imprints fresh bio-ometric information on $UD_{x}$ and computes $(BIO_{UR_{x}}^{n}, RP_{n}) = Gen(BIO_{UR_{x}}^{o}, W_{n} = H(ID_{UR_{x}} \parallel PSW_{UR_{x}}^{n}), P_{n}^{0} = (W_{n}^{o} \oplus W_{n}^{n})$. In addition to this, $UR_{x}$ calculates $AD_{6}^{n} = R_{UR_{x}}^{n}Ke_{n}^{n} = P_{n}^{0} \oplus BIO_{UR_{x}}^{n}$. Moreover, $UR_{x}$ by using ASCON encryption computes $(CT_{UR_{x}}^{n}, Tag_{UR_{x}}^{n}) = E_{Ke_{n}}(\{AD_{6}^{n}, PT_{UR_{x}}^{n}\})$, where $PT_{UR_{x}}^{n} = \{PID, SP_{UR_{x}}^{n}\}$. Finally, $UR_{x}$ updates the credentials $\{CT_{UR_{x}}^{n}, Tag_{UR_{x}}^{n}, Gen(), Rep(), RP, R_{UR_{x}}^{n}\}$ with $\{CT_{UR_{x}}^{0}, Tag_{UR_{x}}^{0}, Gen(), Rep(), RP, R_{UR_{x}}^{0}\}$ in its own memory.

V. SECURITY ANALYSIS

In this section, the resiliency of the proposed REAS-TMIS against various security treats is demonstrated by conducting informal analysis and SK security is proved through ROM based formal security analysis. In addition to this, the security of REAS-TMIS is illustrated through Scyther-based validation.

A. INFORMAL SECURITY ANALYSIS

This subsection demonstrates the informal security analysis of REAS-TMIS scheme, to show its resistance against various security attacks.

1) SMCL ATTACK

If $A$ obtained the smart device or card of $UR_{x}$. Then $UR_{x}$ can procure the sensitive information, such as $\{CT_{UR_{x}}^{n}, Tag_{UR_{x}}^{n}, Gen(), Rep(), RP, R_{UR_{x}}^{n}\}$, stored in the memory of $UD_{x}$ or smart card and can perform various attacks on behalf of $UR_{x}$. However, the information stored in the memory of $UR_{x}$ in encrypted form and $A$ can not extract any useful information, such as $\{PSW_{UR_{x}}, ID_{UR_{x}}, BIO_{UR_{x}}\}$ from the encrypted information to launch an attack. Therefore REAS-TMIS is capable of thwarting SMCL attack.

2) PGU/PUD ATTACK

In this attack, the objective of $A$, after retrieving the critical information, i.e., $\{CT_{UR_{x}}, Tag_{UR_{x}}^{n}, Gen(), Rep(), RP, R_{UR_{x}}^{n}\}$, is to update the secret credentials, such as $\{PSW_{UR_{x}}, ID_{UR_{x}}, BIO_{UR_{x}}\}$. For this, $A$ picks the random credentials, such as $\{PSW_{UR_{x}}^{A}, ID_{UR_{x}}^{A}, BIO_{UR_{x}}^{A}\}$ and computes
M. Tanveer et al.: REAS-TMIS: Resource-Efficient Authentication Scheme for Telecare Medical Information System

| User $UR_x$ | Medical Server $MS$ |
|-------------|---------------------|
| $(CTx_{UR_x}, Tag_{UR_x}, Gen(\cdot), Rep(\cdot), RU_{UR_x}, RP)$ | $(SID, SP_{UR_x})$ |

inputs password $PSW_{UR_x}$ and identity $ID_{UR_x}$, imprints bio-metric $BI_{UR_x}$ and computes $BIK_{UR_x} = Rep(BIO_{UR_x}, RP)$, $P_r = H(ID_{UR_x} \parallel PSW_{UR_x})$, $AD_1 = RU_{UR_x}$, $K_{e1} = (P^e_{\text{e1}} \oplus P^b_{\text{e1}}) \oplus BIK_{UR_x}$, $(PT_{UR_x}, Tag_{UR_x}) = BKE_{K_{e1}}(\{AD_1\}, CT_{UR_x})$, verifies the condition $Tag_{UR_x} = Tag_{UR_x}$, if holds, picks $RA_x$, $R_{T1}$ and $CT_{1}$ and computes $P_3 = H(PID \parallel RA_x \parallel TS_1)$, $AD_2 = (P^e_{\text{e2}} \oplus P^b_{\text{e2}})$, $K_{e2} = (P^e_{\text{e2}} \oplus P^b_{\text{e2}})$, $(CT_1, Tag_1) = E_{K_{e2}}(\{AD_2\}, RA_x)$, $MES_r:(TS_1, PID, CT_1, Tag_2, R_{T1})$, $(UR_x \rightarrow MS)$.

checks $Td > |Tr_x - TS_1|$, if so, computes $AD_3 = AD_1$, $K_{e3} = (SP_{UR_x} \parallel R_{T1})$, $(PT_{MS}, Tag_2) = E_{K_{e3}}(\{AD_3\}, CT_2)$, $Tag_2 = Tag_2$, if holds, $PT_{MS} = (PID^{\text{new}} \parallel R_x \parallel SK_V)$, $SK_{UR_x} = H(P_3 \parallel P_1 \parallel RA_x \parallel PID^{\text{new}})$, computes $SK_V = (SK^a_{UR_x} \oplus SK^b_{UR_x})$, checks $SK_V = SK_{UR_x}$, updates $PID$ with $PID^{\text{new}}$, determines $PT_{MS}^{\text{new}} = (PID^{\text{new}} \parallel SP_{UR_x})$, $(CT_{UR_x}, Tag_{UR_x}^{\text{new}}) = E_{K_{e3}}(\{AD_3\}, PT_{MS}^{\text{new}})$, updates $(R_{T2_x}, CT_{UR_x}, Tag_{UR_x}^{\text{new}})$ with $(R_{T2_x}^{\text{new}} \parallel CT_{UR_x}^{\text{new}} \parallel Tag_{UR_x}^{\text{new}})$, where $AD_4 = R_{T2_x}^{\text{new}}$. Finally, $UD_x$ checks the condition $Tag_{UR_x}^{\text{new}} = Tag_{UR_x}$.

4) REPLAY ATTACK

As described in Sections IV-C, during the AKE process, the exchanged messages incorporate the latest current timestamps. During the AKE phase, the exchanged message procuring entities verify the timestamp received with the messages to guarantee it is not greater than the allowed time delay $Td$. Therefore, REAS-TMIS is resistant to replay attack.

5) MATM ATTACK

To effectuate MATM attack, $A$ expropriates the message $MES_1 : \{TS_1, PID, CT_1, Tag_1, R_{T1}\}$ communicated during the AKE process. $A$ then generates messages with modified parameters, such as $MES_1' : \{TS'_1, PID', CT'_1, Tag'_1, R_{T1}'\}$ and disseminates the $MES_1'$ to $MS$. After receiving $MES_1'$ checks the condition $Tag'_1 = Tag_2$ to ensure the authenticity of the received $MES_1'$. This will not hold because it is hard for $A$ to generate a valid message on behalf of $UR_x$ without knowing its secret credentials $P_1$ and $SP_{UR_x}$. Thus, REAS-TMIS is resilient against MATM attack.
6) DoS ATTACK
In the proposed REAS-TMIS, UR, can send the AKE request to MS after achieving the local authentication. Local authentication phase prevents UR from sending a large volume of AKE request to MS to overwhelm the message processing resources of MS. So, in REAS-TMIS, UD, checks the condition TagUR ≡ TagUR to accomplish local authentication. In this way, REAS-TMIS is capable of resisting DoS attack.

7) IMPERSONATION ATTACK
To deploy URIM attack, A captures the message MES1 : \{TS1, PID, CT1, Tag1, Rb\} disseminated during the AKE process and fabricates MES1', which is a modified message. A then disseminates the MES1' to MS to make believe MS that MES1' is from a legitimate entity of the network. However, A cannot succeed in generating a legist MES1', out with knowing the secret credential P1 and SPUR. In addition, A succeeds in generating MES2 : \{TS2, CT2, Tag3\} without knowing the secret credentials P1, Ra, MKMS, and SPUR. Thus, REAS-TMIS is resilient against URIM and SIM attacks.

8) EPLE ATTACK
In the proposed REAS-TMIS, SK is construed as SKUR, = SKMS = H(P3 || P1 || Ra || Re || PIDnew), where P3 = H(PID || Rb || TS1), P1 = (W1 + W2) and PIDnew = (TIDUR) || NP2 || MP2. It is obvious, that SKUR, = SKMS is constructed using ephemeral secrets (ES), such as (Rb, Re, NP) and long-term secrets (LTS), such as \{P1, MKMS, SPUR\}. Therefore, to compromise SK, A requires to know both ES and LTS. Thus, REAS-TMIS is resistant to EPLE and SIM attacks.

B. ROM-BASED FORMAL SECURITY ANALYSIS
This section renders the ROM-based analysis of the proposed REAS-TMIS protocol to verify SK’s security, established between UR, and MS. Under the ROM, the security of the proposed REAS-TMIS is given in Theorem 1. According to the ROM of the REAS-TMIS the \(t^0\) instance of an entity \(Ψ\) is denoted by \(Ψ^0\). Moreover, UR and MS are denoted as the entities \(Ψ^0_{UR}\) and \(Ψ^0_{MS}\), and their \(t_1^0\) and \(t_2^0\) instances are represented as \(Ψ^1_{UR}\) and \(Ψ^2_{MS}\) respectively. The hash function (SHA-256) is irreversible and collision resistant, which is modeled as random oracle \(Shash\). Moreover, the ROM describes the queries Tabulated in Table 3, which are utilized by A to simulate an attack.

Definition 1: Let A is polynomial time plt adversary running against the AED scheme and effectuates \(Q\) queries of length \(lth\), then A’s online chosen ciphertext attack (OCCAS) advantage can be described as follows [50]–[52]:

\[
\text{Ad}_{A}^{\text{OCCAS}}(\text{plt}) \leq \text{Ad}_{A}^{\text{OPRP-CPA}}(\text{Ques, lth, plt}) + \text{Ad}_{A}^{\text{INT-CTXT}}(\text{Ques, lth, plt}),
\]

where \(\text{Ad}_{A}^{\text{INT-CTXT}}\) signifies A’s advantage on integrity of the ciphertext and \(\text{Ad}_{A}^{\text{OPRP-CPA}}\) denotes A’s advantage on online pseudo-random permutation chosen-plaintext attack.

Theorem 1: Let A running against REAS-TMIS in plt to acquire the constructed SK, established between UR, and MS during the AKE phase. A’s advantage to break SK’s security can be defined as follows

\[
\text{Ad}_{A}^{\text{REAS-TMIS}}(\text{plt}) \leq \frac{\text{HSQ}^2_{\text{que}}}{\text{Shash}} + \frac{\text{SQ}^2_{\text{que}}}{2} + \text{Ad}_{A}^{\text{OCCAS}}(\text{plt}),
\]

where \(|\text{PSD}|\), \(\text{HSQ}^2_{\text{que}}, \text{Shash}, \text{SQ}^2_{\text{que}}, \text{Ad}_{A}^{\text{OCCAS}}(\text{plt})\) denote the password dictionary, hash queries, output size of hash function, send queries, and A’s advantage on an AED scheme.

Proof: We define the following five games \((Gm_h|h = 0, 1, 2, 3, 4)\) to establish the proof of theorem 1. In addition A’s advantage in breaking SK’s security is represented as \(\text{Ad}_{A}^{\text{REAS-TMIS}} = |2 \cdot \text{Pr}[\text{SC}] - 1|\), where \(\text{Pr}[\text{SC}]\) represents an event, in which A wins by guessing the correct bit B in \(Gm_h\). Under ROM, REAS-TMIS is contemplated as protected if \(\text{Ad}_{A}^{\text{REAS-TMIS}}\) is insignificant.

Gm0: Under ROM, in this game, an actual attack is launched by A against the proposed REAS-TMIS. A at the starting of \(Gm_0\) selects “bit B”. Thus, the following can be achieved

\[
\text{Ad}_{A}^{\text{REAS-TMIS}}(\text{plt}) = |2 \cdot \text{Ad}_{A}^{\text{REAS-TMIS}} - 1|.
\]

Gm1: This models the eavesdropping attack, wherein A expropriates the messages, i.e, MES1 : \{TS1, PID, CT1, Tag1, Rb\} and MES2 : \{TS2, CT2, Tag3\}, which are communicated via the public communication channel using \(\text{Execute}(Ψ^0_{UR}, Ψ^3_{MS})\) query. After expropriating MES1 and MES2, A attempts to constructs the session key and performs \(\text{Reveal}\) and \(\text{Test}\) to validate whether the constructed key is real key or a random number. As discussed in the Section IV-C, the established SK \(SK_{UR,} = SK_MS = H(P3 || P1 || Ra || Re || PID_{new})\) is constructed by utilizing ES \{Ra, Rb\} and LTS \{P1, IDMS, IDUR, SPUR\}, which are unknown to A. Therefore, A derive SK. Thus, \(Gm_0\) and \(Gm_1\) are indistinguishable and following can be achieved.

\[
\text{Ad}_{A}^{\text{REAS-TMIS}} = \text{Ad}_{A}^{\text{REAS-TMIS}}.
\]

Gm2 : By the simulating the Hash oracle, A attempts to effectuate an active attack. During the AKE process, MES1 incorporates PID = (TIDUR) || NP || MP, which is protected by \(MP = H(ID_MS || MK_MS)\) and MP is protected by hash function (SHA-256). As the hash function is irreversible and collision resistant. Thus, A cannot extract the sensitive parameter P1 from PID. Therefore, by birthday paradox, we can deduce

\[
|\text{Ad}_{A}^{\text{REAS-TMIS}} - \text{Ad}_{A}^{\text{REAS-TMIS}}| \leq \frac{\text{HSQ}^2_{\text{que}}}{2} ||\text{Shash}||.
\]
A
eries are encrypted with ASCON, which is an AEAD scheme.

which are used to construct SK. However, these secret param-
eries are difficult to guess. By the definition (1), we can

By using (38) and (40), we obtain

|Ad^REAS–TMIS_{A,Gm3} − Ad^REAS–TMIS_{A,Gm4} | ≤ SQ_{que}^{2lbbk−1|PSD|}. (42)

By using (35), (36), (37), and (42), we get

Ad^REAS–TMIS_{A}(plt) ≤ \frac{HSQ_{stu}^{2}}{|Shash|} + \frac{SQ_{que}^{2lbbk−1|PSD|}}{2} + 2 \cdot Ad^{OCCA}_{ASCON,A}(plt). \quad (43)

C. SCYTHER BASED FORMAL SECURITY VERIFICATION

Scyther [53] is a python-based software tool used to verify the security of a security scheme. We use the Scyther tool to validate the security robustness of REAS-TMIS against various covert and pernicious security threats. Scyther can identify different security lapses efficiently. Scyther has found its footprints in the extensive utilization in validating and analyzing AKE schemes or security protocols. Scyther presented superior performance contrasted to existing tools employed to verify AKE schemes’ security.

REAS-TMIS is coded in Scyther utilizing Security Protocol Description Language (SPDL). In the SPDL script, there are two roles defined, i.e., (i) UR (user role) and (ii) MS (server role). In addition, we define the claims, such as claim(UR, Secret, SK) and claim(MS, Secret, SK) manually., which are validated by the Scyther as shown in Fig. 3. Moreover, the claims generated automatically by Scyther, such as for the user role, the claim(UR, Alive), claim(UR, Weakagree), claim(UR, Niaagree), and claim (UR, Nisynch) are verified. Moreover, for MS role, the claim (MS, Alive), claim(MS, Weakagree), claim(MS, Niaagree), and claim (MS, Nisynch) are also validated by Scyther as shown in Fig. 3. Therefore, REAS-TMIS is secure against various malicious security threats.

VI. PERFORMANCE EVALUATION

To evaluate the effectiveness and efficiency of the proposed REAS-TMIS, we compare it with the related AKE schemes in terms of security functionalities and computational, communication, and storage overheads. The related AKE schemes include the scheme of Qui et al. [18], Kumari et al. [6], Mo et al. [35], Arshad et al. [54], and Ostad et al. [55]. In addition to this, to simulate UD_A by utilizing PA attack. However, in REAS-TMIS, the stored information are in the encrypted form and encryption is performed using the credentials \{PSW_{UR}, ID_{UR}, BIK_{UR}\}, where BIK_{UR} (barometric key) is difficult to guess and generate. Thus, without the knowledge of valid credentials \{PSW_{UR}, ID_{UR}, BIK_{UR}\}, it is impractical for A to extract the secret credentials used in the AKE process. Moreover, the length of the bio-metric key is \frac{1}{2^{200}}, where lbbk denotes the length of bio-metric key. Therefore, the probability of guessing BIK_{UR} is negligible. In addition to this, only a limited number wrong password attempts are allowed. Under these conditions, following can be deducted

\begin{align*}
|Ad^REAS–TMIS_{A,Gm2} − Ad^REAS–TMIS_{A,Gm3} | ≤ \frac{SQ_{que}^{2lbbk−1|PSD|}}{2}. \quad (36)
\end{align*}

\begin{align*}
Gm4 : In Gm4, A launches an active attack against by eavesdropping the exchanged messages, such as MES_{1} : \{TS_{1}, PID, CT_{1}, Tag_{1}, R_{1}\} and MES_{2} : \{TS_{2}, CT_{2}, Tag_{2}\}. After capturing MES_{1} and MES_{2}, A to extract the secret parameter, which are used to construct SK. However, these secret parameters are encrypted with ASCON, which is an AEAD scheme. Therefore, A cannot extract the secret credential form the encrypted information. Thus, by the definition (1), we can deduced
\begin{align*}
|Ad^REAS–TMIS_{A,Gm3} − Ad^REAS–TMIS_{A,Gm4} | ≤ Ad^{OCCA}_{ASCON,A}(plt). \quad (37)
\end{align*}

To this end, all the relevant queries associated with the above Gm4 are accomplished. The only event is left to imagine the arbitrary bit B after accomplishing the Reveal and Test queries. Consequently, we have

\begin{align*}
Ad^REAS–TMIS_{A,Gm4} = \frac{1}{2}. \quad (38)
\end{align*}

From (33) and (34), we get
\begin{align*}
Ad^REAS–TMIS_{A}(plt) = |2 \cdot Ad^REAS–TMIS_{A,Gm3} − 1|. \quad (39)
\end{align*}

From (39), we get

\begin{align*}
|2 \cdot Ad^REAS–TMIS_{A}(plt) = |Ad^REAS–TMIS_{A,Gm3} − \frac{1}{2}|. \quad (40)
\end{align*}

By using (38) and (40), we obtain
\begin{align*}
\frac{1}{2} \cdot Ad^REAS–TMIS_{A}(plt) = |Ad^REAS–TMIS_{A,Gm3} − Ad^REAS–TMIS_{A,Gm4} | \quad (41)
\end{align*}

By employing triangular inequality, we get
\begin{align*}
\frac{1}{2} \cdot Ad^REAS–TMIS_{A}(plt) \leq |Ad^REAS–TMIS_{A,Gm3} − Ad^REAS–TMIS_{A,Gm4} | + |Ad^REAS–TMIS_{A,Gm2} − Ad^REAS–TMIS_{A,Gm3} |.
\end{align*}

\begin{table}[h]
\centering
\caption{Explanation of Various ROM Queries.}
\begin{tabular}{|c|c|}
\hline
Query & Purpose \tabularnewline \hline
Test(0) & This query executed by A to verify whether device B is secret or not by sending a valid device ID or not. \tabularnewline \hline
Reveal(0) & This query executed by A to reveal SK. \tabularnewline \hline
Capture(0) & This query executed by A to capture exchanged messages between U_R and MS. \tabularnewline \hline
Receive(0) & This query executed by A to receive the secret credential form the U_R. \tabularnewline \hline
\end{tabular}
\end{table}
et al. [6], Mo et al. [35], Arshad et al. [54], and Ostad et al. [55] require \(8T_{HS} + 2T_{ECC} \approx 8.46\) ms, \(12T_{HS} + 3T_{ECC} + 2T_{ENC} \approx 13.47\) ms, \(7T_{HS} + 3T_{ECC} + T_{ECA} \approx 11.09\) ms, \(8T_{HS} + 2T_{ECC} + T_M \approx 8.52\) ms, and \(11T_{HS} + 2T_{ECC} + 2T_{ECA} \approx 9.74\) ms, respectively, at UD, which are 48.23%, 67.48%, 60.5%, 48.59%, and 55.03%, respectively, higher than REAS-TMIS. Moreover, the computational overhead required by REAS-TMIS during the AKE process at MS side is \(3T_{HS} + 2T_{AS} \approx 0.38\) ms. Conversely, the computational overhead required by the schemes of Qui et al. [18], Kumari et al. [6], Mo et al. [35], Arshad et al. [54], and Ostad et al. [55] are \(5T_{HS} + 2T_{ECC} \approx 1.81\) ms, \(9T_{HS} + 3T_{ECC} + 2T_{ENC} \approx 2.82\) ms, \(6T_{HS} + 3T_{ECC} + T_{ECA} \approx 2.64\) ms, \(8T_{HS} + 2T_{ECC} + T_M + T_{inv} \approx 1.96\) ms, and \(8T_{HS} + 2T_{ECC} + 2T_{ECA} + T_{ECA} \approx 2.0\) ms, respectively, at MS, which are 79.01%, 86.52%, 85.61%, 80.61%, and 81%, respectively, higher than REAS-TMIS. Table 4 illustrates that the total computational overhead required by REAS-TMIS to accomplish the AKE process is 4.72 ms, which is 54.04%, 71.03%, 65.62%, 55.18%, 59.8% lower than the related AKE schemes. Furthermore, Fig 5 shows that the computational overhead increases when the number of users increases.

### B. COMPUTATIONAL OVERHEAD

To compute the computational overhead required to accomplish the AKE process, we contemplate the computational complexities of various cryptographic primitive presented in Table 5. The computational overhead at the user device (UD) side needed to accomplish the AKE phase is \(3T_{HS} + 4T_{AS} + T_B \approx 4.38\) ms, while the schemes of Qui et al. [18], Kumari et al. [6], Mo et al. [35], Arshad et al. [54], and Ostad et al. [55] are 4.46 ms, 8 ms, 38 ms, \(81\) ms, and 82 ms, respectively, at MS, with 96 ms, and 81 ms, respectively, higher than REAS-TMIS. Figure 4 shows that the computational overhead increases when the number of users increases.

### C. COMMUNICATION OVERHEAD

To estimate the communication overhead, we consider the size of random numbers, Tag, AD PID, hash function output, timestamps, and ECC point 128, 128, 128, 256, 256, 32, and 320 bits, respectively. During the AKE phase of REAS-TMIS, two messages are communicated, such as \(MES_1 : \{TS_1, PID, CT_1, Tag_1, R_b\}\) and \(MES_2 : \{TS_2, CT_2, Tag_3\}\).
TABLE 6. Security Functionalities/Properties Comparison.

| Features | Kumari et al. [6] | Qui et al. [18] | Mo et al. [35] | Arshad et al. [54] | Ostad et al. [55] | REAS-TMIS |
|----------|-------------------|----------------|----------------|-------------------|------------------|-----------|
| PIN      | ✓                 | ✓              | ✓              | ✓                 | ✓                | ✓         |
| SIM      | ✓                 | ✓              | ✓              | ✓                 | ✓                | ✓         |
| PGU      | ×                 | ×              | ✓              | ✓                 | ✓                | ✓         |
| URM      | ×                 | ×              | ✓              | ✓                 | ✓                | ✓         |
| URA      | ×                 | ×              | ✓              | ✓                 | ✓                | ✓         |
| SMCL     | ×                 | ×              | ✓              | ✓                 | ✓                | ✓         |
| MATM     | ✓                 | ✓              | ✓              | ✓                 | ✓                | ✓         |
| D-SYN    | ×                 | ×              | ✓              | ✓                 | ✓                | ✓         |
| DoS      | ✓                 | ✓              | ✓              | ✓                 | ✓                | ✓         |
| RA       | ✓                 | ✓              | ✓              | ✓                 | ✓                | ✓         |
| EPLE     | ✓                 | ✓              | ✓              | ✓                 | ✓                | ✓         |
| ROM      | ✓                 | ✓              | ✓              | ✓                 | ✓                | ✓         |
| SV       | ✓                 | ✓              | ✓              | ✓                 | ✓                | ✓         |
| KCA      | ✓                 | ✓              | ✓              | ✓                 | ✓                | ✓         |

KCA: Key compromised attack; ✓: Represents the available functionality; ×: Represents the functionality is not applicable.

FIGURE 5. The computational cost rises at MS with increasing authentication requests from URx.

TABLE 7. Comparative Analysis Computational Time Required to Accomplish AKE Phase.

| Protocol/Scheme | Total Computational Time |
|-----------------|--------------------------|
| Qui et al. [18] | 13T_h + 4T ECC ≈ 10.27 ms |
| Kumari et al. [6] | 21T_h + 6T ECC + 4T ENC ≈ 16.29 ms |
| Mo et al. [35] | 13T_h + 6T ECC + 2T ENC ≈ 13.73 ms |
| Arshad et al. [54] | 16T_h + 4T ECC + 2T + T ENC ≈ 10.53 ms |
| Ostad et al. [55] | 19T_h + 4T ECC + 4T ENC + 2T ENC ≈ 11.74 ms |
| REAS-TMIS | 8T_h + 6T ENC + T ENC ≈ 4.72 ms |

with size {32 + 256 + 128 + 128 + 16} = 560 bits and {32 + 512 + 128} = 672 bits, respectively. Total communication overhead is [560 + 672] = 1232 bits. The scheme of Qui et al. [18], Kumari et al. [6], Mo et al. [35], Arshad et al. [54], and Ostad et al. [55] require 1536 bits, 1628 bits, 1674 bits, 2462 bits, and 1696 bits, respectively, which are 19.79%, 24.32%, 26.4%, 49.96%, and 27.36% higher than REAS-TMIS. A comparative analysis of the communication overhead between REAS-TMIS and related AKE schemes is given in Table 8.

D. STORAGE OVERHEAD COMPARISON

In the proposed scheme, URx and MS require storing \{ CT_{URx}, Tag_{URx}, Gen(·), Rep(·), RP, R_{URx} \} and \{ SID, SP_{URx} \} of size (256 + 128 + 128 + 160 + 128) = 800 bits and (128 + 128) = 256 bits, respectively. Total memory requirement of REAS-TMIS is (800 + 256) = 1056 bits. The scheme of Qui et al. [18], Kumari et al. [6], Mo et al. [35], Arshad et al. [54], and Ostad et al. [55] require storing 768 bits, 1312 bits, 1632 bits, 1660 bits, and 1920 bits, respectively. Storage overhead comparison is given in Fig 6. REAS-TMIS require slightly more memory requirements than Qui et al. [18]. However, REAS-TMIS provides enhanced security functionalities and requires low computational and communication overheads than the scheme of Qui et al. [18].

FIGURE 6. Comparative analysis of storage overhead.
VII. CONCLUSION

Security and privacy are imperative for critical environments like TMIS where sensitive information is communicated through the public Internet. In this paper, we have used AEAD and hash function and proposed an AKE scheme, called REAS-TMIS, for the TMIS environment that enables users to efficiently make authentication and establish SK with MS. REAS-TMIS is computationally inexpensive and fitting for resource-constrained IoT devices in TMIS. Moreover, the scheme enables doctors and nurses to securely access the information stored at MS. Aside from this, we have formally proved the security of the SK through the ROM. Moreover, we have also proved, through informal analysis, the strength of the scheme against various security attacks, such as replay, impersonation, and DoS attacks. Additionally, we have executed Scyther-based formal security analysis and have showed the security strength of the scheme. Moreover, we have also proved, through informal analysis, the strength of the scheme against various security attacks, such as replay, impersonation, and DoS attacks. Additionally, we have executed Scyther-based formal security analysis and have showed the security strength of the scheme. Moreover, we have also proved, through informal analysis, the strength of the scheme against various security attacks, such as replay, impersonation, and DoS attacks. Additionally, we have executed Scyther-based formal security analysis and have showed the security strength of the scheme.

In conclusion, the developed scheme of REAS-TMIS is relatively more secure than the related AKE scheme. Moreover, a comparison with the state-of-the-art is presented to have showed the security strength of the scheme. More-
[36] S. Barman, A. K. Das, D. Samanta, S. Chattopadhyay, J. J. P. C. Rodrigues, and Y. Park, “Provably secure multi-server authentication protocol using fuzzy commitment,” IEEE Access, vol. 6, pp. 38578–38594, 2018.

[37] R. Amin, T. Maitra, D. Giri, and P. D. Srivastava, “Cryptanalysis and improvement of an RSA based remote user authentication scheme using smart card,” Wireless Pers. Commun., vol. 96, no. 3, pp. 4629–4659, Oct. 2017.

[38] M. Nikooghadam, R. Jahantigh, and H. Arshad, “A lightweight authentication and key agreement protocol preserving user anonymity,” Multimedia Tools Appl., vol. 76, no. 11, pp. 13401–13423, 2017.

[39] R. Ali and A. K. Pal, “An efficient three factor-based authentication scheme in multiserver environment using ECC,” Int. J. Commun. Syst., vol. 31, no. 4, pp. e3484, Mar. 2018.

[40] F. Wang, G. Xu, C. Wang, and J. Peng, “A provably secure biometrics-based authentication scheme for multiserver environment,” Secur. Commun. Netw., vol. 2019, pp. 1–15, Jun. 2019.

[41] S. A. Chaudhry, “Combating identity de-synchronization: An improved lightweight symmetric key based authentication scheme for IoV,” J. Netw. Comput. Appl., vol. 125, Feb. 2021, Art. no. 106529.

[42] D. Dolev and A. C. Yao, “On the security of public key protocols,” IEEE Trans. Inf. Theory, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.

[43] M. Tanveer, A. U. Khan, H. Shah, S. A. Chaudhry, and M. Ahmad, “ARAP-SG: Anonymous and reliable authentication protocol for smart grids,” IEEE Access, vol. 9, pp. 143366–143377, 2021.

[44] S. A. Chaudhry, “Correcting ‘PALK: Password-based anonymous lightweight key agreement framework for smart grid,’” Int. J. Electr. Power Energy Syst., vol. 125, Feb. 2021, Art. no. 106529.

[45] R. Canetti and H. Krawczyk, “Analysis of key-exchange protocols and key agreement schemes,” in Proc. Int. Conf. Comput. Aided Verification, Berlin, Germany: Springer, 2001, pp. 453–474.

[46] C. Dobrzaunig, M. Eichlseder, F. Mendel, and M. Schläffer (2014), Ascon, Submission to the CAESAR Competition. [Online]. Available: http://ascon.iak.tugraz.at

[47] C. Beierle, A. Biryukov, L. C. dos Santos, J. Grootschädl, L. Perrin, A. Udovenko, V. Velichkov, and Q. Wang, “Lightweight AEAD and hashing using the sparkle permutation family,” IACR Trans. Symmetric Cryptol., vol. 2020, no. 1, pp. 208–261, Jun. 2020.

[48] F. Abed, C. Forler, and S. Lucks, “General classification of the authenticated encryption schemes for the CAESAR competition,” Comput. Sci. Rev., vol. 22, pp. 13–26, Nov. 2016.

[49] M. Tanveer, A. U. Khan, N. Kumar, and M. M. Hassan, “RAMP-IoD: A robust authenticated key management protocol for the Internet of Drones,” IEEE Internet Things J., vol. 9, no. 2, pp. 1353–1355, Jan. 2022.

[50] M. Tanveer, G. Abbas, Z. H. Abbas, M. Bilal, A. Makherjee, and K. S. Kwak, “LAKE-6SH: Lightweight user authenticated key exchange for 6LoWPAN-based smart homes,” IEEE Internet Things J., vol. 9, no. 4, pp. 2578–2591, Feb. 2022.

[51] C. J. Cremers, “The scytter tool: Verification, falsification, and analysis of security protocols,” in Proc. Int. Conf. Comput. Aided Verification. Berlin, Germany: Springer, 2008, pp. 414–418.

[52] H. Arshad and M. Nikooghadam, “Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems,” J. Med. Syst., vol. 38, no. 12, pp. 1–12, Dec. 2014.

[53] A. Ostad-Sharif, D. Abbasinezhad-Mood, and M. Nikooghadam, “A robust and efficient ECC-based mutual authentication and session key generation scheme for healthcare applications,” J. Med. Syst., vol. 43, no. 1, pp. 1–22, Jan. 2019.

ABD ULLAH KHAN (Member, IEEE) received the B.S. degree (Hons.) in telecommunication from the UST Bannu, in 2013, the M.Sc. degree in electrical engineering from COMSATS University Islamabad, in 2016, and the Ph.D. degree in computer science from the Gulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan, in 2021. Part of his Ph.D. degree is from the National University of Sciences and Technology, Islamabad. He is currently working as an Assistant Professor with the National University of Science and Technology, Balochistan Campus, Quetta, Pakistan. His research interests include resource allocation and management in wireless networks, artificial intelligence, and network security. He was a recipient of the prestigious scholarship of the Higher Education Commission of Pakistan. He has received a Gold Medal for his B.S. degree. Besides, he is an Active Reviewer of IEEE Network, IEEE INTERNET OF THINGS JOURNAL, IEEE SYSTEMS JOURNAL, IEEE ACCESS, and Computer Communications (Elsevier).

AHMED ALKHAYYAT (Member, IEEE) received the B.Sc. degree in electrical engineering from Al Kufa University, Najaf, Iraq, in 2007, the M.Sc. degree from the Dehradun Institute of Technology, Dehradun, India, in 2010, and the Ph.D. degree from Cankaya University, Ankara, Turkey, in 2015. He is currently the Dean of International Relationship and a Manager of the word ranking with Islamic University, Najaf. His research interests include the IoT in the health-care systems, SDN, network coding, cognitive radio, efficient-energy routing protocols, and efficient-energy MAC protocol in cooperative wireless networks and wireless body area networks, as well as cross-tier designing for self-organized networks. He contributed in organizing a several IEEE conferences, workshop, and special sessions. To serve his community, he acted as a reviewer for several conferences and journals.

SHEHZAD ASHRAF CHAUDHRY received the master’s and Ph.D. degrees (Hons.). Before this, he worked as an Associate Professor of computer science with the University of Sialkot, and International Islamic University, Islamabad, Pakistan. He is currently an Associate Professor with the Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelsim University, Istanbul, Turkey. He is working in the field of information and communication security, he has published extensively in prestigious venues, such as IEEE Communication Standards Magazine, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, the IEEE TRANSACTIONS ON RELIABILITY, ACM TRANSACTIONS ON INTERNET TECHNOLOGY, Sustainable Cities and Society (Elsevier), FGCS, IIEPES, and Computer Networks. Over 125 publications and with an H-index of 34, i-10 index of 72, and accumulate impact factor of more than 300, he has published more than 100 SCI/E indexed manuscripts and has been cited more than 3200 times. He has also supervised more than 40 graduate students in their research. In 2018, considering his research, Pakistan Council for Science and Technology granted him the Prestigious Research Productivity Award, while affirming him among Top Productive Computer Scientist in Pakistan. Recently, he is listed among Top 2% computer scientists across the world in Stanford University’s report. His current research interests include lightweight cryptography, elliptic/hyper elliptic curve cryptography, multimedia security, e-payment systems, MANETs, SIP authentication, smart grid security, IP multimedia subsystems, and next generation networks. He occasionally writes on issues of higher education in Pakistan. He was awarded gold medal for achieving maximum distinction of 4/4 CGPA in his master’s degree.

MUHAMMAD TANVEER received the M.S. degree in computer science from GC University, Lahore, Pakistan, in 2017, and the Ph.D. degree in computer science from the GIK Institute of Engineering Sciences and Technology, Pakistan, in 2022. He is currently acting as an Assistant Professor at the University of Management and Technology, Lahore. His research interests include communication networks and network security.
YOUSAF BIN ZIKRIA (Senior Member, IEEE) is currently working as an Assistant Professor with the Department of Information and Communication Engineering, Yeungnam University, South Korea. He authored more than 100 refereed articles, conference papers, book chapters, and patents. His journal article’s cumulative impact factor (IF) is more than 300. He published articles at the top venue, including IEEE COMMUNICATIONS, SURVEYS, AND TUTORIALS, IEEE Wireless Communications Magazine, IEEE NETWORK, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, Future Generation Computer Systems (Elsevier), Sustainable Cities and Society (Elsevier), and Journal of Network and Computer Applications (Elsevier). He has managed numerous FT/SI in SCI/E indexed journals. His research interests include the IoT, 5G, machine learning, wireless communications and networks, WSNs, routing protocols, CRAHN, CRASN, transport protocols, VANETS, embedded systems, and network and information security. He also held the prestigious CISA, JNCIS-SEC, JNCIS-ER, JNCIA-ER, JNCIA-EX, and Advance Routing Switching and WAN Technologies certifications. He has been listed in the world’s top 2% of researchers published by Elsevier and Stanford University.

SUNG WON KIM received the B.S. and M.S. degrees from the Department of Control and Instrumentation Engineering, Seoul National University, South Korea, in 1990 and 1992, respectively, and the Ph.D. degree from the School of Electrical Engineering and Computer Sciences, Seoul National University, in August 2002. From January 1992 to August 2001, he was a Researcher at the Research and Development Center, LG Electronics, South Korea. From August 2001 to August 2003, he was a Researcher at the Research and Development Center of AL Tech, South Korea. From August 2003 to February 2005, he was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. In March 2005, he joined the Department of Information and Communication Engineering, Yeungnam University, Gyeongsangbuk-do, South Korea, where he is currently a Professor. His research interests include resource management, wireless networks, mobile computing, performance evaluation, and machine learning.