This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3153069, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

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

MUHAMMAD TANVEER[1], ABD ULLAH KHAN[2], AHMED ALKHAYYAT[3], SHEHZAD ASHRAF CHAUDHRY[4], YOUSAF BIN ZIKRIA[5] (Senior Member, IEEE), SUNG WON KIM[5]

[1]Muhammad Tanveer is with Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology, Topi 23640, Pakistan. (e-mail: tanveer123giki@gmail.com).

[2]Abd Ullah Khan is with the School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan, (e-mail: abdullah@nbc.nust.edu.pk).

[3]Ahmed Alkhayyat is with Department of Computer Technical Engineering, College of Technical Engineering, Islamic University, Najaf 54001, Iraq. (e-mail: ahmedalkhayyat85@iunajaf.edu.iq).

[4] Shehzad Ashraf Chaudhry is with Department of Computer Engineering, Istanbul Gelisim University, Istanbul, Turkey. (e-mail: sashraf@gelisim.edu.tr)

[5]Yousaf Bin Zikria and Sung Won Kim are with the Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Korea (e-mail: yousafbinzikria@ynu.ac.kr, swon@yu.ac.kr)

Corresponding author: Yousaf Bin Zikria (e-mail: yousafbinzikria@ynu.ac.kr) and Sung Won Kim (swon@yu.ac.kr).

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3153069, IEEE Access

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 constricted 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

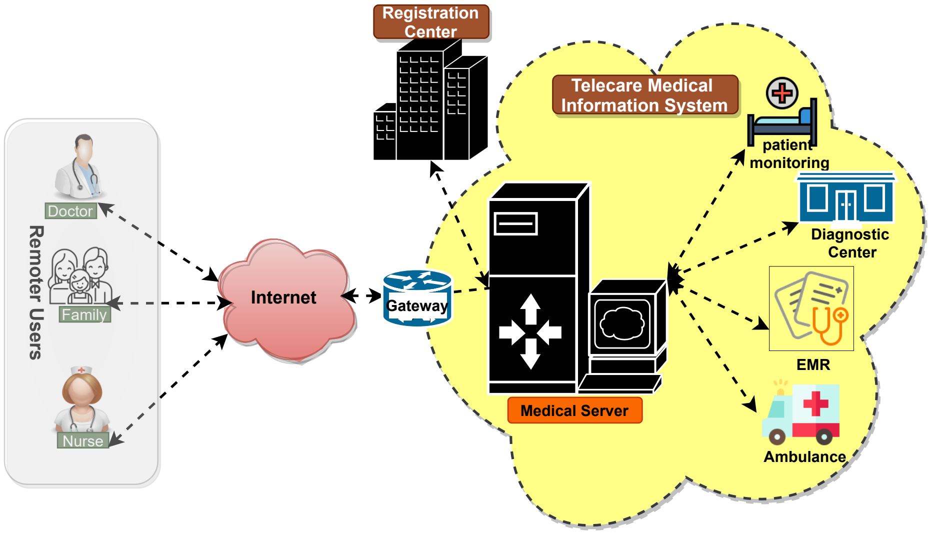

Figure 1: Telecare medical information system.

Table 1: User Authentication Schemes: A detailed summary

Protocol	Limitations	Operation Utilized
Kumari *et al.* [6]	Unable to impede PGU, SMCL, URA, PIN, URIM, and D-SYN attack	ECC, Exclusive-OR, and SHA
Jia *et al.* [17]	Does not render resistance against EPLE attack.	ECC, Exclusive-OR, and SHA
Qui *et al.* [18]	Unable to impede URIM attacks. In addition, unable to provide URA feature.	ECC, Exclusive-OR, and SHA
Son *et al.* [19]	Unable to prevent secret disclosure and PIN attacks.	BP, Exclusive-OR, and SHA
Zhang *et al.* [20]	Unable to impede PGU and SIM attacks.	Exclusive-OR, chaotic map, SHA
He *et al.* [21]	Cannot thwart DoS, PIN, and PGU attacks.	BP, Exclusive-OR, and SHA
Alzahrani *et al.* [22]	Unable to impede the SIM attack.	Exclusive-OR and SHA
Khatoon *et al.* [7]	Incapable of thwarting URIM and PIN attacks and cannot provide URA feature.	BP, Exclusive-OR, and SHA
Tanveer *et al.* [23]	Unable to impede D-SYN and PIN attacks.	ASCON, Exclusive-OR, and SHA
Han *et al.* [24]	Does not prevent MATM, URA, URIM, and SIM attacks.	ECC, SHA, and Exclusive-OR
Nayak *et al.* [25]	Unable to impede D-SYN and cannot render URA feature.	SHA and Exclusive-OR
Chaudhry *et al.* [26]	Cannot restrain impersonation and EPLE attacks and cannot render anonymity feature.	ECC, SHA, and Exclusive-OR
Li *et al.* [8]	Unable to impede PGU, IDGU, URIM, PIN, and SMCL attacks.	ECC, SHA, and Exclusive-OR
Das *et al.* [9]	Cannot thwart SIM, MATM, URIM, and PIN attacks. Unable to provide URA property.	Exclusive-OR and SHA
Mir *et al.* [27]	Does not thwart PIN and MATM attacks.	Exclusive-OR and SHA
Amin *et al.* [28]	Unable to impede replay, PGU, and URIM attacks.	SHA and Exclusive-OR
Madhusudhan *et al.* [10]	Cannot resist replay, MATM, PIN, and SIM. Does not provide MA and URA features.	Chaotic map, Exclusive-OR, and SHA
Renuka *et al.* [29]	Cannot resist PIN and provide URA feature	ECC, SHA, and Exclusive-OR
Dharminder *et al.* [30]	Unable to impede PIN attack.	RSA, SHA, and Exclusive-OR
Shen *et al.* [31]	Incapable of preventing SMCL, DoS, and PFS attacks.	Exclusive-OR, ECC, and SHA
Chaudhry *et al.* [32]	Incapable of preventing SMCL, DoS, and PFS attacks.	Exclusive-OR, AES, and SHA
Irshad *et al.* [33]	Does not render revocation feature.	Exclusive-OR, ECC, and SHA
Irshad *et al.* [34]	Cannot not render resistance against EPLE and SIM attacks.	Exclusive-OR, ECC, and SHA
Mo *et al.* [35]	Cannot render protection replay, stolen verifier, DoS, and EPLE attacks.	Exclusive-OR, ECC, and SHA
Barman *et al.* [36]	Incapable of preventing PIN attack.	Exclusive-OR, ECC, and SHA
Amin *et al.* [37]	Incapable of preventing URIM and PGU attacks and does not ensure PFS feature.	Exclusive-OR, RSA, and SHA
Nikooghadam *et al.* [38]	Cannot resist SIM, URIM, PGU, SK disclosure, and D-SYN attacks and does not provide PFS feature.	Exclusive-OR, AES, and SHA
Ali and Pal *et al.* [39]	Unable to impede PIN, SIM, URIM, and DoS attacks.	Exclusive-OR, ECC, and SHA
Wang *et al.* [40]	Cannot resist URIM, SIM, and EPLE attacks.	Exclusive-OR, ECC, and SHA

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 $(UR_x|x = 1, 2, 3, \cdots, K)$, where K is the number of users. The users can be doctors, nurses, or family members authorized to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3153069, IEEE Access

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 UR_x. It is often the case that UR_x requires the data/information stored at MS. Thus, a security mechanism is required to enable safe communication between UR_x and MS. To provide UR_x with secure access to the system resources, an AKE scheme is required.

B. ATTACK MODEL

The Dolev–Yao (DY) [44], [45] is considered as the threat model (TM) 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 [46]. 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 [47]. It is an effective TM than the DY model and is considered a substantially acceptable model for devising an AKE scheme.

III. PRELIMINARIES

This section renders the background knowledge of different preliminaries utilized in devising the REAS-TMIS.

A. FUZZY EXTRACTOR

Fuzzy extractor (FE) is used to derive a unique secret key from the user's bio-metric template. FE comprises bio-metric key generation and reproduction functions, denoted by $Gen(\cdot)$ and $Rep(\cdot)$, respectively. The function $Gen(\cdot)$ is a probabilistic function and takes input of user bio-metric (BIO_{UR}) information and generates a unique key $BIK \in [0,1]^{lbbk}$ and RP , where $lbbk$ is the length of BIK and RP is the reproduction parameter. Moreover, $Rep(\cdot)$ deterministic function and reproduces BIK by taking $BIO_{UR}^{'}$ and RP as the inputs with the condition $(BIO_{UR}, BIO_{UR}^{'} \leq Et)$, where $BIO_{UR}^{'}$ and Et are the bio-metric information entered at the time of login and error tolerance.

B. ASCON

ASCON [48] is an online AEAD scheme, which provides the confidentiality, integrity, and authentic of the data simultaneously. The encryption and decryption processes of the ASCON can be expressed as follows.

$$(CT, Tag) = \mathcal{E}_{Ke}\{(IV, AD), PT\} \quad (1)$$

$$(PT, Tag^{'}) = \mathcal{D}_{Ke}\{(IV, AD), CT\} \quad (2)$$

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

Table 2: List of Notations Used in REAS-TMIS

Notation	Description
$\mathcal{A}$	Adversary
ID_{MS}, MK_{MS}	Real identity and master key of the medical server (MS)
PSW_{UR_x}, ID_{UR_x}	Real identity and password of user
PID	Pseudo identity of user of size 256 bits
SP_{RU_x}	Secret parameter of user of size 128 bits
SID	Searching identity of size 128 bits
NP, NP_2	Random numbers, each of size 128 bits
TS_1 , TS_2	Timestamps used during the AKE process each of size 32 bits
(CT_x, Tag_x)	Ciphertext and Tag generated by the encryption process of an AEAD, where $x = 1, 2, 3, 4$
$AD_1, AD_2, AD_3, AD_4, AD_5$	Associative data used in the encryption process, each of size 128 bits
R_a, R_c	Random numbers used in the AKE process, each of size 128 bits
R_b	Random number used in the AKE process of size 16 bits
$\mathcal{E}_{Ke}(xy)$	Represents the encryption of message "xy" with key Ke
$\mathcal{D}_{Ke}(xy)$	Represents the decryption of message "xy" with key Ke
Ke_x	Secret key used in the encryption process, where $x = l, 1, 2, 3, 4$
BIO_{UR_x}, RP	Bio-metric information and reproduction parameter
BIK_{UR_x}	128 bits bio-metric key generated by FE
$Rep(\cdot)$	Bio-metric key reproduction function of FE
$Gen(\cdot)$	Bio-metric key generation function of FE
RP	Reproduction parameter generated by $Gen(\cdot)$
$H(\cdot)$	Collision resistant hash function
$\oplus, \|$	Exclusive-OR and concatenation operations

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, UR_x needs to register with RC. RC assigns secret credentials to UR_x during the URR phase. Before accessing the network resources UR_x needs to authenticate itself with MS. RC accomplishes the following imperative steps to register UR_x.

1) Step URR-1

UR_x picks its own identity ID_{UR_x}, random number R_{UR_x}, and password PSW_{UR_x}. In addition, UR_x by employing

FE computes $(BIK_{UR_x}, RP) = Gen(BIO_{UR_x})$, where the size of BIK_{UR_x} is 128 bits. Moreover, UR_x calculates $W_1 = H(ID_{UR_x} \parallel PSW_{UR_x})$ and $P_1 = (W_1^a \oplus W_2^b)$ 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_1^a and W_2^b, 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 $MP_1 = H(ID_{MS} \parallel MK_{MS})$, $TID_{UR_x} = (P_1 \oplus NP)$, $PID = (TID_{UR_x} \parallel NP) \oplus MP_1$, $W_2 = H(P_1 \parallel MK_{MS})$, and searching identity $SID = (W_2^a \oplus W_2^b)$. Moreover, RC computes $W_3 = H(MK_{MS} \parallel P_1 \parallel ID_{MS})$ and secret parameter $SP_{UR_x}{=}(W_3^a \oplus W_3^b)$ and sends $\{PID$, $SP_{UR_x}\}$ to UR_x. In addition, RC, stores the credentials $\{SID$, $SP_{UR_x}\}$ in MS's database.

3) Step URR-3

On procuring the secret credentials form RC, UR_x computes $AD_1 = R_{UR_x}$ and $Ke_l = P_1 \oplus BIK_{UR_x}$. Moreover, UR_x by using ASCON encryption computes $(CT_{UR_x}, Tag_{UR_x}) = \mathcal{E}_{Ke_l}$ $\{(AD_1)$, $PT_{UR_x}\}$, where $PT_{UR_x} = \{PID$, $SP_{UR_x}\}$. Finally, UR_x stores the parameters $\{CT_{UR_x}$, Tag_{UR_x}, $Gen(\cdot)$, $Rep(\cdot)$, RP, $R_{UR_x}\}$ in its own memory.

C. AKE PHASE

In this phase, UR_x performs the local authentication by validating its secret credentials and then sends the AKE request to MS. After achieving the mutual authentication both UR_x 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 $PSW_{UR_x}^l$, and identity $ID_{UR_x}^l$, imprints bio-metric BIO_{UR_x} and computes the followings

$$BIK_{UR_x}^l = Rep(BIO_{UR_x}^l, RP), \tag{3}$$

$$P_l = H(ID_{UR_x}^l \parallel PSW_{UR_x}^l) \; and \; AD_l = R_{UR_x} \tag{4}$$

$$Ke_l = (P_l^a \oplus P_l^b) \oplus BIK_{UR_x}^l, \tag{5}$$

$$(PT_{UR_x}, Tag_{UR_x}^l) = \mathcal{D}_{Ke_l}\{(AD_l), CT_{UR_x}\}, \tag{6}$$

where $BIK_{UR_x}^l$ is the bio-metric key of associated with UR_x, which is obtained by using $Rep(\cdot)$ function of of FE. The parameter P_1 is determined by performing hash operation on ID_{UR_x} and PSW_{UR_x}. Moreover, the secret encryption key Ke_l is determined by concatenating $(P_l^a \oplus P_l^b)$ and $BIK_{UR_x}^l$, where P_l^a and P_l^b are derived from P_l. Furthermore, PT_{UR_x} and $Tag_{UR_x}^l$ are the output of the ASCON decryption algorithm. Finally, smart user device UD_x checks the the following condition

$$Tag_{UR_x}^l \stackrel{?}{=} Tag_{UR_x}. \tag{7}$$

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

$$P_3 = H(PID \parallel R_b \parallel TS_1) \tag{8}$$

$$AD_2 = (P_3^a \oplus P_3^b), \tag{9}$$

$$Ke_1 = (P_l^a \oplus P_l^b), \tag{10}$$

$$(CT_1, Tag_1) = \mathcal{E}_{Ke_1}\{(AD_2), R_a\}, \tag{11}$$

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 $Td \geq |Tre - TS_1|$, where Td 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

$$MP_2 = H(ID_{MS} \parallel MK_{MS}), \tag{12}$$

$$MP_2 \oplus PID = (TID_{UR_x} \parallel NP), \tag{13}$$

$$P_1 = (TID_{UR_x} \oplus NP) \tag{14}$$

$$Ke_2 = (P_1^a \oplus P_1^b), \tag{15}$$

$$P_3 = H(PID \parallel R_b \parallel TS_1), \tag{16}$$

$$AD_3 = (P_3^a \oplus P_3^b), \tag{17}$$

$$(R_a, Tag_2) = \mathcal{D}_{Ke_2}\{(AD_3), CT_1\}, \tag{18}$$

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 $Tag_1 \stackrel{?}{=} Tag_2$, if holds, MS retrieves R_a. Moreover, MS computes

$$Q_5 = H(P_1 \parallel MK_{MS}) \; and \; SID = (Q_5^a \oplus Q_5^b). \tag{19}$$

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_5^a and Q_5^b , which are two parts of Q_5. SID is used to retrieves the secret parameter SP_{UR_x}

IEEE Access

from the database of MS. Furthermore, MS picks TS_2, R_c, and NP_2, and computes

$$QM = H(MK_{MS} \parallel P_1 \parallel ID_{MS}), \tag{20}$$

$$Ke_3 = (QM^a \oplus QM^b) \oplus R_a, \tag{21}$$

$$TID^n_{UR_x} = (P_1 \oplus NP_2), \tag{22}$$

$$PID^{new} = (TID^n_{UR_x} \parallel NP_2) \oplus MP_2, \tag{23}$$

$$SK_{MS} = H(P_3 \parallel P_1 \parallel R_a \parallel R_c \parallel PID^{new}), \tag{24}$$

$$SKv = (SK^a_{MS} \oplus SK^b_{MS}), \tag{25}$$

$$PT_{MS} = (PID^{new} \parallel R_c \parallel SKv), \tag{26}$$

$$(CT_2, Tag_3) = \mathcal{E}_{Ke_3}\{(AD_4), PT_{MS}\}, \tag{27}$$

where Ke_3 is the secret key which is used in the encryption process, which is derived by splitting QM into two parts. PID^{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 the session key, which is used to ensure the encrypted communication with UR_x. SKv 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_c, PID^{new}, and SKv. 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 $Td \geq |Tre - TS_2|$. If MES_2 is fresh, UD_x determines $AD_5 = AD_1$ and $Ke_4 = (SP_{UR_x} \oplus R_a)$. In addition, By using ASCON decryption algorithm, UD_x computes

$$(PT_{MS}, Tag_4) = \mathcal{D}_{Ke_4}\{(AD_5), CT_2\} \tag{28}$$

and checks the condition $Tag_4 \stackrel{?}{=} Tag_3$, if it holds, UD_x considers MES_2 as a valid message and procures $PT_{MS} = (PID^{new} \parallel R_C \parallel SKv)$, which is the plaintext. Moreover to ensure the encrypted communication with MS, UD_x computes SK as follows

$$SK_{UR_x} = H(P_3 \parallel P_1 \parallel R_a \parallel R_c \parallel PID^{new}). \tag{29}$$

Furthermore, UD_x computes $SKv_1 = (SK^a_{UR_x} \oplus SK^b_{UR_x})$ and checks the condition $SKv \stackrel{?}{=} SKv_1$. If the condition is satisfied, UD_x updates PID with PID^{new} and determines $PT^{new}_{UR_x} = \{PID^{new}, SP_{UR_x}\}$. In addition to this, UD_x picks $R^{new}_{UR_x}$ and computes

$$(CT^{new}_{UR_x}, Tag^{new}_{UR_x}) = \mathcal{E}_{Ke_l}\{(AD_6), PT^{new}_{UR_x}\}. \tag{30}$$

where $AD_6 = R^{new}_{UR_x}$. Finally, UD_x updates $\{R_{UR_x}$, CT_{UR_x}, $Tag_{UR_x}\}$ with $\{R^{new}_{UR_x}$, $CT^{new}_{UR_x}$, $Tag^{new}_{UR_x}\}$ 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 $P_1 = H(ID_{UR_x} \parallel PSW^o_{UR_x})$ and sends P_1 to RC. RC derives SID as $SID = (W^a_2 \oplus W^b_2)$. 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 $PSW^o_{UR_x}$ and ID_{UR_x} and imprints its bio-metric information $BIO^o_{UR_x}$ at the available interface of UD_x. Moreover, UD_x computes $BIK^o_{UR_x} = Rep(BIO^o_{UR_x}, RP)$, $P_o = H(ID_{UR_x} \parallel PSW^o_{UR_x})$, $AD^o_l = R^o_{UR_x}$ $Ke_o = (P^a_o \oplus P^b_o) \oplus BIK^o_{UR_x}$, $(PT_{UR_x}, Tag^o_{UR_x}) = \mathcal{D}_{Ke_o}$ $\{(AD^o_l), CT_{UR_x}\}$, Finally, UD_x validates condition $Tag^o_{UR_x} \stackrel{?}{=} Tag_{UR_x}$. If its holds, UD_x sends a message to UR_x to enter new secret credentials.

2) Step PUD-2

UR_x picks new random number $R^n_{UR_x}$ and password $PSW^n_{UR_x}$. In addition, UR_x imprints fresh bio-metric information on UD_x and computes $(BIK^n_{UR_x}, RP^n) = Gen(BIO^n_{UR_x})$, $W_n = H(ID_{UR_x} \parallel PSW^n_{UR_x})$, and $P^n_1 = (W^a_n \oplus W^b_n)$. In addition to this, UR_x calculates $AD^n_1 = R^n_{UR_x}$ and $Ke^n_l = P^n_1 \oplus BIK^n_{UR_x}$. Moreover, UR_x by using ASCON encryption computes $(CT^n_{UR_x}, Tag^n_{UR_x}) = \mathcal{E}_{Ke^n_l}$ $\{(AD^n_1), PT^n_{UR_x}\}$, where $PT^n_{UR_x} = \{PID, SP_{UR_x}\}$. Finally, UR_x updates the credentials $\{CT_{UR_x}$, Tag_{UR_x}, $Gen(\cdot)$, $Rep(\cdot)$, RP, $R_{UR_x}\}$ with $\{CT^n_{UR_x}$, $Tag^n_{UR_x}$, $Gen(\cdot)$, $Rep(\cdot)$, RP^n, $R^n_{UR_x}\}$ 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.

User UR_x	**Medical Server** MS
$\{CTx_{UR_x}, Tag_{UR_x}, Gen(\cdot), Rep(\cdot), R_{UR_x}, RP\}$	$\{ SID, SP_{UR_x}\}$
inputs password $PSW^l_{UR_x}$, and identity $ID^l_{UR_x}$, imprints bio-metric BIO_{UR_x} and computes $BIK^l_{UR_x} = Rep(BIO^l_{UR_x}, RP)$, $P_l = H(ID_{UR_x} \parallel PSW_{UR_x})$, $AD_l = R_{UR_x}$, $Ke_l = (P^a_l \oplus P^b_l) \oplus BIK^l_{UR_x}$, $(PT_{UR_x}, Tag^l_{UR_x}) \stackrel{?}{=} \mathcal{D}_{Ke_l}\{(AD_l), CT_{UR_x}\}$, verifies the condition $Tag^l_{UR_x} = Tag_{UR_x}$, if holds, picks R_a, R_b, and TS_1 and computes $P_3 = H(PID \parallel R_b \parallel TS_1)$, $AD_2 = (P^a_3 \oplus P^b_3)$, $Ke_1 = (P^a_l \oplus P^b_l)$, $(CT_1, Tag_1) = \mathcal{E}_{Ke_1}\{(AD_2), R_a\}$, $\xrightarrow[(UR_x \rightarrow MS)]{MES_1:\{TS_1,\ PID,\ CT_1,\ Tag_1,\ R_b\}}$. checks $Td \geq \lvert Tre - TS_2\rvert$, if so, computes $AD_5 = AD_1$, $Ke_4 = (SP_{UR_x} \oplus R_a)$, $(PT_{MS}, Tag_4) = \mathcal{D}_{Ke_4}\{(AD_5), CT_2\}$, $Tag_4 \stackrel{?}{=} Tag_3$, if it holds, $PT_{MS} = (PID^{new} \parallel R_c \parallel SKv)$, $SK_{UR_x} = H(P_3 \parallel P_1 \parallel R_a \parallel R_c \parallel PID^{new})$, computes $SKv_1 = (SK^a_{UR_x} \oplus SK^b_{UR_x})$, checks $SKv \stackrel{?}{=} SKv_1$, updates PID with PID^{new}, determines $PT^{new}_{UR_x} = \{PID^{new}, SP_{UR_x}\}$, $(CT^{new}_{UR_x}, Tag^{new}_{UR_x}) = \mathcal{E}_{Ke_l}\{(AD_6), PT^{new}_{UR_x}\}$ updates $\{R_{UR_x}, CT_{UR_x}, Tag_{UR_x}\}$ with $\{R^{new}_{UR_x}, CT^{new}_{UR_x}, Tag^{new}_{UR_x}\}$, where $AD_6 = R^{new}_{UR_x}$.	checks $Td \geq \lvert Tre - TS_1\rvert$, if holds, computes $MP_2 = H(ID_{MS} \parallel MK_{MS})$, $MP_2 \oplus PID = (TID_{UR_x} \parallel NP)$, $P_1 = (TID_{UR_x} \oplus NP)$, $Ke_2 = (P^a_1 \oplus P^b_1)$, $P_3 = H(PID \parallel R_b \parallel TS_1)$, $AD_3 = (P^a_3 \oplus P^b_3)$, $(R_a, Tag_2) = \mathcal{D}_{Ke_2}\{(AD_3), CT_1\}$, validates $Tag_1 \stackrel{?}{=} Tag_2$, if holds, $Q_5 = H(P_1 \parallel MK_{MS})$, $SID = (Q^a_5 \oplus Q^b_5)$, retrieves SP_{UR_x} related to SID, picks TS_2, R_c, and NP_2, computes $QM = H(MK_{MS} \parallel P_1 \parallel ID_{MS})$, $Ke_3 = (QM^a \oplus QM^b) \oplus R_a$, $TID^n_{UR_x} = (P_1 \oplus NP_2)$, $PID^{new} = ((TID^n_{UR_x}) \parallel NP_2) \oplus MP_2$, $SK_{MS} = H(P_3 \parallel P_1 \parallel R_a \parallel R_c \parallel PID^{new})$, $SKv = (SK^a_{MS} \oplus SK^b_{MS})$, $AD_3 = AD_4$, $PT_{MS} = (PID^{new} \parallel R_c \parallel SKv)$, $(CT_2, Tag_3) = \mathcal{E}_{Ke_3}\{(AD_4), PT_{MS}\}$, $\xleftarrow[(MS \leftarrow UR_x)]{MES_2:\{TS_2,\ CT_2,\ Tag_3\}}$.
$SK_{UR_x} (= SK_{MS}) = H(P_3 \parallel P_1 \parallel R_a \parallel R_c \parallel PID^{new})$	

Figure 2: Authentication phase of REAS-TMIS.

1) SMCL Attack

If $\mathcal{A}$ obtained the smart device or card of UR_x. Then UR_x can procure the sensitive information, such as $\{CT_{UR_x}$, $Tag_{UR_x}, Gen(\cdot), Rep(\cdot), RP, R_{UR_x}\}$, 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 $\mathcal{A}$ can not extract ant 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 $\mathcal{A}$, after retrieving the critical information,i.e, $\{CT_{UR_x}$, Tag_{UR_x}, $Gen(\cdot)$, $Rep(\cdot)$, RP, $R_{UR_x}\}$, is to update the secret credentials, such as $\{PSW_{UR_x}, ID_{UR_x}, BIO_{UR_x}\}$. For this, $\mathcal{A}$ picks the random credentials, such as $\{PSW^{\mathcal{A}}_{UR_x}, ID^{\mathcal{A}}_{UR_x}, BIO_{UR_{\mathcal{A}}}\}$ and computes $BIK^{\mathcal{A}}_{UR_x} = Rep(BIO^{\mathcal{A}}_{UR_x}, RP)$, $P_o = H(ID^{\mathcal{A}}_{UR_x} \parallel PSW^{\mathcal{A}}_{UR_x})$, $AD^{\mathcal{A}}_l = R^{\mathcal{A}}_{UR_x}$ $Ke_{\mathcal{A}} = (P^a_{\mathcal{A}} \oplus P^b_{\mathcal{A}}) \oplus BIK^{\mathcal{A}}_{UR_x}$, and $(PT_{UR_x}, Tag^{\mathcal{A}}_{UR_x})$= $\mathcal{D}_{Ke_o}$ $\{(AD^{\mathcal{A}}_l)$, $CT_{UR_x}\}$. Finally, UD_x checks the condition $Tag^{\mathcal{A}}_{UR_x} \stackrel{?}{=} Tag_{UR_x}$. However, $\mathcal{A}$ cannot not perform the decryption process without knowing the valid secret credentials of UR_x. In addition, the bio-metric keys are difficult to predict/generate or guess. Therefore, REAS-TMIS is capable of resisting PGU/PUD attack.

3) Anonymity and Untraceability

REAS-TMIS ensure the anonymity of entities of the network. There are two messages exchanged, i.e, MES_1 : $\{TS_1$, PID, CT_1, Tag_1, $R_b\}$ and MES_2 :$\{TS_2$, CT_2, $Tag_3\}$ to complete the AKE process. After capturing MES_1 and MES_2, $\mathcal{A}$ cannot extract the real identity of UR_x from PID = $((TID_{UR_x}) \parallel NP) \oplus MP$. Thus, REAS-TMIS capable of resisting IDGU attack. In addition, MES_1 and MES_2 are dynamic as they are generated using random numbers and current timestamps. Hence, $\mathcal{A}$ cannot generate correlate the messages captured from two different AKE sessions. Therefore, REAS-TMIS ensure the URA and untraceablity features.

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3153069, IEEE Access

delay Td . Therefore, REAS-TMIS is resistant to replay attack.

5) MATM Attack

To effectuate MATM attack, $\mathcal{A}$ expropriates the message MES_1 : $\{TS_1,\ PID,\ CT_1,\ Tag_1,\ R_b\}$ communicated during the AKE process. $\mathcal{A}$ then generates messages with modified parameters, such as $MES_1^{'}$: $\{TS_1^{'},\ PID^{'},\ CT_1^{'},\ Tag_1^{'},\ R_b^{'}\}$ and disseminates the $MES_1^{'}$ to MS. After receiving $MES_1^{'}$ checks the condition $Tag_1^{'} \stackrel{?}{=} Tag_2$ to ensure the authenticity of the received $MES_1^{'}$. This will not hold because it is hard for $\mathcal{A}$ to generate a valid message on behalf of UR_x without knowing its secret credentials P_1 and SP_{UR_x}. In addition, $\mathcal{A}$ cannot succeed in generating a valid message MES_2 :$\{TS_2, CT_2, Tag_3\}$ without knowing the secret credentials of MS, such as P_1, R_a, MK_{MS}, and SP_{UR_x}. Thus, REAS-TMIS is resilient against MATM attack.

6) DoS Attack

In the proposed REAS-TMIS, UR_x can send the AKE request to MS after achieving the local authentication. Local authentication phase prevents UR_x from sending a large volume of AKE request to MS to overwhelm the message processing resources of MS. So, in REAS-TMIS, UD_x checks the condition $Tag^l_{UR_x} \stackrel{?}{=} Tag_{UR_x}$ to accomplish local authentication. In this way, REAS-TMIS is capable of resisting DoS attack.

7) Impersonation Attack

To deploy URIM attack, $\mathcal{A}$ captures the message MES_1 : $\{TS_1, PID, CT_1, Tag_1, R_b\}$ disseminated during the AKE process and fabricates $MES_1^{'}$, which is a modified message. $\mathcal{A}$ then disseminates the $MES_1^{'}$ to MS to make believe MS that $MES_1^{'}$ is from a legitimate entity of the network. However, $\mathcal{A}$ cannot succeed in generating a licit MES_1 with out knowing the secret credential P_1 and SP_{UR_x}. In addition, $\mathcal{A}$ succeed in generating MES_2 :$\{TS_2,\ CT_2,\ Tag_3\}$ without knowing the secret credentials P_1, R_a, MK_{MS}, and SP_{UR_x}. Thus, REAS-TMIS is resilient against URIM and SIM attacks.

8) EPLE Attack

In the proposed REAS-TMIS, SK is construed as $SK_{UR_x}(= SK_{MS}) = H(P_3 \parallel P_1 \parallel R_a \parallel R_c \parallel PID^{new})$, where P_3 $H(PID \parallel R_b \parallel TS_1)$, $P_1 = (W_1^a \oplus W_2^b)$, and $PID^{new} = ((TID^n_{UR_x}) \parallel NP_2) \oplus MP_2$. It is obvious, that $SK_{UR_x}(= SK_{MS})$ is constructed using ephemeral secrets (ES), such as$\{R_a,\ R_c,\ R_b,\ NP\}$ and long-term secrets (LTS), such as $\{P_1,\ MK_{MS},\ SP_{UR_x}\}$. Therefore, to compromise SK, $\mathcal{A}$ requires to know both ES and LTS. Thus, REAS-TMIS is resistant to EPLE attack.

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_x 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^{th} instance of an entity Ψ is denoted by Ψ^p. Moreover, UR_x and MS are denoted as the entities Ψ_{UR_x} and Ψ_{MS}, and their t_1^{th} and t_2^{th} instances are represented as $\Psi^{p1}_{UR_x}$ and Ψ^{p2}_{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 $\mathcal{A}$ to simulate an attack.

Definition 1. *Let $\mathcal{A}$ is polynomial time plt adversary running against the AEAD scheme and effectuates Que queries of length lth, then $\mathcal{A}$'s online chosen ciphertext attack (OCCA3) advantage can be described as follows [50]–[52].*

$$Ad^{OCCA3}_{\varphi,\mathcal{A}}(plt) \leq Ad^{OPRP-CPA}_{\varphi}(Que, lth, plt) + Ad^{INT-CTXT}_{\varphi}(Que, lth, plt), \quad (31)$$

where $Ad^{INT-CTXT}_{\varphi}$ signifies $\mathcal{A}$'s advantage on integrity of the ciphertext and $Ad^{OPRP-CPA}_{\varphi}$ denotes $\mathcal{A}$'s advantage on online pseudo-random permutation chosen-plaintext attack.

Theorem 1. *Let $\mathcal{A}$ running against REAS-TMIS in plt to acquire the constructed SK, established between UR_x and MS during the AKE phase. $\mathcal{A}$'s advantage to break SK's security can be defined as follows*

$$Ad^{REAS-TMIS}_{\mathcal{A}}(plt) \leq \frac{HSQ^2_{que}}{|Shash|} + \frac{SQ_{que}}{2^{lbbk-1}|PSD|} + 2 \cdot Ad^{OCCA3}_{ASCON,\mathcal{A}}(plt), \quad (32)$$

where $|PSD|$, HSQ_{que}, $|Shash|$, SQ_{que}, and $Ad^{OCCA3}_{\mathcal{A}}(plt)$ denote the password dictionary, hash queries, output size of hash function, send queries, and $\mathcal{A}$'s advantage on an AEAD 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 $\mathcal{A}$'s advantage in breaking SK's security is represented as $Ad^{REAS-TMIS}_{\mathcal{A},Gm_k} = |2 \cdot Pr[sc] - 1|$, where "$Pr[sc]$" represents an event, in which $\mathcal{A}$ wins by guessing the correct bit B in Gm_h. Under ROM, REAS-TMIS is contemplated as protected if $Ad^{REAS-TMIS}_{\mathcal{A}}$ is insignificant.

Gm_0 : Under ROM, in this game, an actual attack is launched by $\mathcal{A}$ against the proposed REAS-TMIS. $\mathcal{A}$ at the starting of Gm_0 selects "bit B". Thus, the following can be achieved

$$Ad^{REAS-TMIS}_{\mathcal{A}}(plt) = |2 \cdot Ad^{REAS-TMIS}_{\mathcal{A},Gm_0} - 1|. \quad (33)$$

Gm_1 : This models the eavesdropping attack, wherein $\mathcal{A}$ expropriates the messages, i.e, MES_1 : $\{TS_1,\ PID,\ CT_1, Tag_1, R_b\}$ and MES_2 :$\{TS_2, CT_2, Tag_3\}$, which are communicate via the public communication channel using $Execute(\Psi^{p2}_{UR_x}, \Psi^{t3}_{MS})$ query. After expropriating MES_1 and MES_2, **A** attempts to constructs the session key and performs $Reveal$ and $Test$ to validate whether the constructed key is real key or a random number. As discussed in the

Table 3: Explanation of Various ROM Queries

Query	Purpose
$Send(\Psi^{p}, Msg)$	This query executed by $\mathcal{A}$ to generate an active attack by sending a message Msg to Ψ^{p1} and Ψ^{p2} also respond to the received message accordingly.
$Test(\Psi^{p})$	This query executed by $\mathcal{A}$ to verify whether derived SK is correct or random output jut like the outcome of a flipped coin, say B.
$Reveal(\Psi^{p})$	This query executed by $\mathcal{A}$ to procure SK, established to ensure the indecipherable communication between Ψ^{p1} and its associated entity.
$CorruptUR(\Psi^{p1}_{UR_x})$	This query executed by $\mathcal{A}$ to procure secret/sensitive credentials stored in the memory of UR_x by using PA attack.
$Execute(\Psi^{p1}_{UR_x}, \Psi^{p2}_{MS})$	This query executed by $\mathcal{A}$ to capture all the disseminated messages between UR_x and MS.

Section IV-C, the established SK $SK_{UR_x}(= SK_{MS}) = H(P_3 \parallel P_1 \parallel R_a \parallel R_c \parallel PID^{new})$ is constructed by utilizing ES $\{R_a, R_b\}$ and LTS $\{P_1, ID_{MS}, ID_{UR_x}, SP_{UR_x}\}$, which are unknown to $\mathcal{A}$. Therefore, $\mathcal{A}$ derive SK. Thus, Gm_0 and Gm_1 are indistinguishable and following can be achieved.

$$Ad^{REAS-TMIS}_{\mathcal{A},Gm_0} = Ad^{REAS-TMIS}_{\mathcal{A},Gm_1}. \quad (34)$$

Gm_2 : By the simulating the $Hash$ oracle, $\mathcal{A}$ attempts to effectuate an active attack. During the AKE process, MES_1 incorporates $PID = ((TID_{UR_x}) \parallel NP) \oplus MP$, which is protected by $MP = H(ID_{MS} \parallel MK_{MS})$ and MP is protected by hash function (SHA-256). AS the hash function is irreversible and collision resistant. Thus, $\mathcal{A}$ cannot extract the sensitive parameter P_1 from PID. Therefore, by birthday paradox, we can deduce

$$|Ad^{REAS-TMIS}_{\mathcal{A},Gm_1} - Ad^{REAS-TMIS}_{\mathcal{A},Gm_2}| \leq \frac{HSQ^2_{que}}{2|Shash|}. \quad (35)$$

Gm_3 : In Gm_3, $\mathcal{A}$ effectuates an active attack by utilizing the $CorruptUR(\Psi^{p1}_{UR_x})$ query (defined in Table 3). By utilizing this, $\mathcal{A}$ can extricate the information, such as $\{CT_{UR_x}$, Tag_{UR_x}, $Gen(\cdot)$, $Rep(\cdot)$, RP, $R_{UR_x}\}$ stored in the memory of UD_x by utilizing PA attack. However, in REAS-TMIS, the stored information are in the encrypted form and encryption is performed using the credentials $\{PSW_{UR_x}, ID_{UR_x}$ $BIK_{UR_x}\}$, where BIK_{UR_x} (barometric key) is difficult to guess and generate. Thus, without the knowledge of valid credentials $\{PSW_{UR_x}, ID_{UR_x}$ $BIK_{UR_x}\}$, it is impractical for $\mathcal{A}$ to extract the secret credentials used in the AKE process. Moreover, the length of the bio-metric key is $\frac{1}{2^{lbbk}}$, where $lbbk$ denotes the length of bio-metric key. Therefore, the probability of guessing BIK_{UR_x} is negligible. In addition to this, only a limited number wrong password attempts are allowed. Under these, condition following can be deduced

$$|Ad^{REAS-TMIS}_{\mathcal{A},Gm_2} - Ad^{REAS-TMIS}_{\mathcal{A},Gm_3}| \leq \frac{SQ_{que}}{2^{lbbk-1}|PSD|}. \quad (36)$$

Gm_4 : In Gm_4, $\mathcal{A}$ lunches an active attack against by eavesdropping the exchanged messages, such as MES_1 : $\{TS_1$, PID, CT_1, Tag_1, $R_b\}$ and MES_2 :$\{TS_2$, CT_2, $Tag_3\}$. After capturing MES_1 and MES_2, $\mathcal{A}$ 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, $\mathcal{A}$ cannot extract the secret credential form the encrypted information. Thus, by the definition (1), we can deduced

$$|Ad^{REAS-TMIS}_{\mathcal{A},Gm_3} - Ad^{REAS-TMIS}_{\mathcal{A},Gm_4}| \leq Ad^{OCCA3}_{ASCON,\mathcal{A}}(plt). \quad (37)$$

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

$$Ad^{REAS-TMIS}_{\mathcal{A},Gm_4} = \frac{1}{2}. \quad (38)$$

From (33) and (34), we get

$$Ad^{REAS-TMIS}_{\mathcal{A}}(plt) = |2 \cdot Ad^{REAS-TMIS}_{\mathcal{A},Gm_0} - 1|. \quad (39)$$

From (39), we get

$$\frac{1}{2} \cdot Ad^{REAS-TMIS}_{\mathcal{A}}(plt) = |Ad^{REAS-TMIS}_{\mathcal{A},Gm_1} - \frac{1}{2}|. \quad (40)$$

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

$$\frac{1}{2}.Ad^{REAS-TMIS}_{\mathcal{A}}(plt) = |Ad^{REAS-TMIS}_{\mathcal{A},Gm_1} - Ad^{REAS-TMIS}_{\mathcal{A},Gm_4}| \quad (41)$$

By employing triangular inequality, we get

$$\begin{aligned}\frac{1}{2}.Ad^{REAS-TMIS}_{\mathcal{A}}(plt)| \leq{} & |Ad^{REAS-TMIS}_{\mathcal{A},Gm_1} - Ad^{REAS-TMIS}_{\mathcal{A},Gm_2}| \\ & +|Ad^{REAS-TMIS}_{\mathcal{A},Gm_2} - Ad^{REAS-TMIS}_{\mathcal{A},Gm_3}| \\ & +|Ad^{REAS-TMIS}_{\mathcal{A},Gm_3} - Ad^{REAS-TMIS}_{\mathcal{A},Gm_4}|. \end{aligned} \quad (42)$$

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

$$\begin{aligned} Ad^{REAS-TMIS}_{\mathcal{A}}(plt) \leq{} & \frac{HSQ^2_{que}}{|Shash|} + \frac{SQ_{que}}{2^{lbbk-1}|PSD|} \\ & +2 \cdot Ad^{OCCA3}_{ASCON,\mathcal{A}}(plt). \end{aligned} \quad (43)$$

■

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3153069, IEEE Access

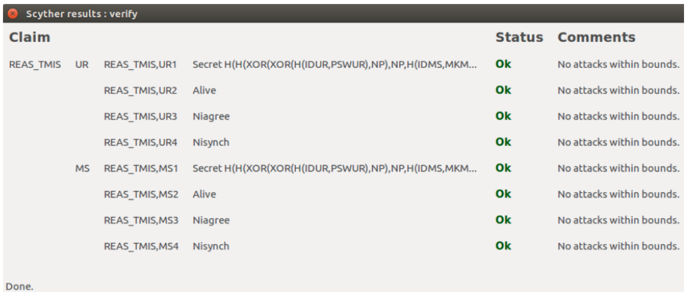

Figure 3: Security analysis of REAS-TMIS using Scyther.

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 $claim(UR, Alive)$, $claim(UR, Weakagree)$, $claim(UR, Niagree)$, and $claim(UR, Nisynch)$ are verified. Moreover, for MS role, the $claim(MS, Alive)$, $claim(MS, Weakagree)$, $claim(MS, Niagree)$, 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_x and MS, we use the platform with specification given in Table 4. Moreover, we use the Python based "PyCrypto" library along with ASCON reference code to obtain the experimental execution time of various cryptographic primitives and ASCON. Table 5 tabulates the experimental computational complexities of different cryptographic primitives.

A. COMPARISON SECURITY FUNCTIONALITIES

An AKE scheme must be secure to impede various security threats. Additionally, an AKE scheme must ensure the anonymity and untraceablity of communicating entities. Table 6 provide a comparative analysis of REAS-TMIS and the related AKE schemes. The scheme of Kumari $et\ al.$ [6] is unable to impede PGU, SMCL, PIN, URIM, and D-SYN attacks and cannot render URA features. The scheme of Qui $et\ al.$ [18] is incapable of impeding URIM attack and unable to provide URA feature. The scheme of Mo $et\ al.$ [35] cannot render protection against replay, stolen verifier, DoS, and EPLE attacks. The scheme of Arshad $et\ al.$ [54] cannot resist replay, MATM, PIN, and SIM attacks. Moreover, the scheme does not provide MA and URA features. The scheme of Ostad $et\ al.$ [55] is incapable of preventing PGU, key compromise, and impersonation attacks. Contrarily, REAS-TMIS is secure and ensures the anonymous communication during the AKE phase.

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_x) 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] 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_x, 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 over head 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 $8T_{HS} + 2T_{ECC} + 2T_{ECA} + 2T_{ENC} \approx 2.0$ 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 increase.

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 message are communicated, such as MES_1 : $\{TS_1$, PID, CT_1, Tag_1, $R_b\}$ and MES_2 :$\{TS_2$, CT_2, $Tag_3\}$ 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

Table 4: Specification of the Platform Used in the experimentation

Device/Platform	Specification
User smart device (UD_x)	Raspberry Pi-3 (R-Pi3) with $\mathcal{CPU}$: Quad core @1.2 GHz, and $\mathcal{RAM}$: 1GB, and $\mathcal{OS}$: Ubuntu 16.04 LTS (64-bit)
MS (server)	Intel(R) Core(TM)/i7-6700 with $\mathcal{CPU}$: @3.40 GHz, $\mathcal{RAM}$: 8 GB and $\mathcal{OS}$: Ubuntu 16.04 LTS (64-bit)

Table 5: Computational Time of Various Cryptographic Operations

Notations	Cryptographic Operation	R-Pi3	MS
T_{ECC}	Computational time of ECC point multiplication	2.850 ms	0.780 ms
T_{ENC}	Computational time of Symmetric encryption	0.391 ms	0.02 ms
T_{ECA}	Computational time of ECC point addition	0.124 ms	.006 ms
T_{inv}	Computational time of modular inversion	-	.0045 ms
T_M	Computational time of modular multiplication	-	0.0025 ms
T_{HS}	Computational time of hash Function (16 bytes)	0.345 ms	0.039 ms
T_{AS}	Computational time of ASCON (AEAD scheme)	0.370 ms	.0351 ms
$T_B \approx T_{ECC}$	Computational time of FE based bio-metric key reproduction	2.850 ms	.780 ms

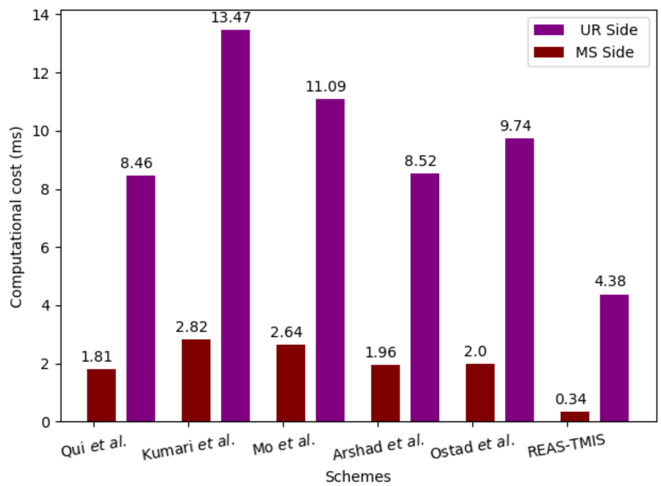

Figure 4: Computational cost at UR_x and MS side.

communication overhead between REAS-TMIS and related AKE schemes is given in Table 8.

D. STORAGE OVERHEAD COMPARISON

In the proposed scheme, UR_x and MS require storing $\{CT_{UR_x}, Tag_{UR_x}, Gen(\cdot), Rep(\cdot), RP, R_{UR_x}\}$ and $\{SID, SP_{UR_x}\}$ 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].

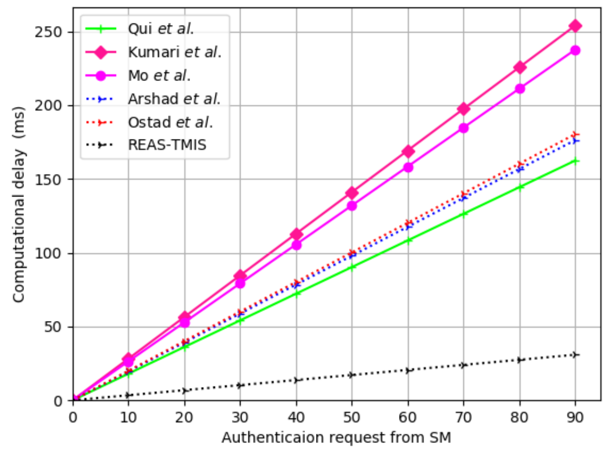

Figure 5: The computational cost rises at MS with increasing authentication requests from UR_x

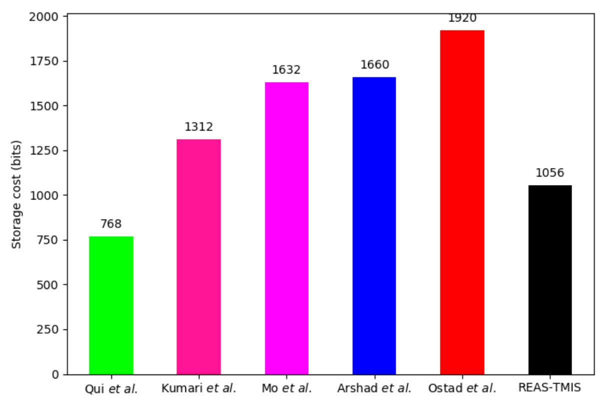

Figure 6: Comparative analysis of storage overhead.

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.

IEEE Access

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	$\times$	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$
SIM	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$
PGU	$\times$	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\checkmark$
URIM	$\times$	$\times$	$\checkmark$	$\checkmark$	$\times$	$\checkmark$
URA	$\times$	$\times$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$
SMCL	$\times$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
MATM	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$
D-SYN	$\times$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
DoS	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\checkmark$
RA	$\checkmark$	$\checkmark$	$\times$	$\times$	$\checkmark$	$\checkmark$
EPLE	$\checkmark$	$\checkmark$	$\times$	$\times$	$\checkmark$	$\checkmark$
ROM	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$
SV	$\checkmark$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\checkmark$
KCA	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\checkmark$

KCA: Key compromised attack; $\checkmark$: Represents the available functionality; $\times$: Represents the functionality is not applicable.

Table 7: Comparative Analysis Computational Time Required to Accomplish AKE Phase

Protocol/Scheme	Total Computational Time
Qui *et al.* [18]	$13T_{HS} + 4T_{ECC} \approx 10.27$ ms
Kumari *et al.* [6]	$21T_{HS} + 6T_{ECC} + 4T_{ENC} \approx 16.29$ ms
Mo *et al.* [35]	$13T_{HS} + 6T_{ECC} + 2T_{ECA} \approx 13.73$ ms
Arshad *et al.* [54]	$16T_{HS} + 4T_{ECC} + 2T + T_{inv} \approx 10.53$ ms
Ostad *et al.* [55]	$19T_{HS} + 4T_{ECC} + 4T_{ECA} + 2T_{ENC} \approx 11.74$ ms
REAS-TMIS	$8T_{HS} + 6T_{AS} + T_B \approx 4.72$ ms

Table 8: Communication Overhead Comparison

AC Protocol	Disseminated Messages During AKE Phase	Total (bits)
Qui *et al.* [18]	$UR_x \xrightarrow{832} MS \xrightarrow{576} UR_x \xrightarrow{256} MS$	**1536**
Kumari *et al.* [6]	$UR_x \xrightarrow{992} MS \xrightarrow{736} UR_x$	**1628**
Mo *et al.* [35]	$UR_x \xrightarrow{842} MS \xrightarrow{576} UR_x \xrightarrow{256} MS$	**1674**
Arshad *et al.* [54]	$UR_x \xrightarrow{1118} MS \xrightarrow{1088} UR_x \xrightarrow{256} MS$	**2462**
Ostad *et al.* [55]	$UR_x \xrightarrow{1120} MS \xrightarrow{576} UR_x$	**1696**
REAS-TMIS	$UR_x \xrightarrow{560} MS \xrightarrow{672} UR_x$	**1232**

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, a comparison with the state-of-the-art is presented to show that REAS-TMIS incurs 54.04% low computational and 19.79% low communication overheads while providing the enhanced security features than the related AKE scheme.

References

[1] M. Al Ameen, J. Liu, and K. Kwak, "Security and privacy issues in wireless sensor networks for healthcare applications," Journal of medical systems, vol. 36, no. 1, pp. 93–101, 2012.

[2] A. U. Khan, G. Abbas, Z. H. Abbas, M. Tanveer, S. Ullah, and A. Naushad, "Hblp: A hybrid underlay-interweave mode crn for the future 5g-based internet of things," IEEE Access, vol. 8, pp. 63 403–63 420, 2020.

[3] J. J. Hathaliya and S. Tanwar, "An exhaustive survey on security and privacy issues in healthcare 4.0," Computer Communications, vol. 153, pp. 311–335, 2020.

[4] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: Challenges and solutions," IEEE Communications Magazine, vol. 55, no. 1, pp. 122–129, 2017.

[5] G. Abbas, M. Tanveer, Z. H. Abbas, M. Waqas, T. Baker, and D. Al-Jumeily OBE, "A secure remote user authentication scheme for 6LoWPAN-based internet of things," PloS one, vol. 16, no. 11, p. e0258279, 2021.

[6] A. Kumari, S. Jangirala, M. Y. Abbasi, V. Kumar, and M. Alam, "ESEAP: ECC based secure and efficient mutual authentication protocol using smart card," Journal of Information Security and Applications, vol. 51, p. 102443, 2020.

[7] S. Khatoon, S. M. M. Rahman, M. Alrubaian, and A. Alamri, "Privacy-preserved, provable secure, mutually authenticated key agreement protocol for healthcare in a smart city environment," IEEE Access, vol. 7, pp. 47 962–47 971, 2019.

[8] X. Li, J. Niu, M. Karuppiah, S. Kumari, and F. Wu, "Secure and efficient two-factor user authentication scheme with user anonymity for network based e-health care applications," Journal of medical systems, vol. 40, no. 12, pp. 1–12, 2016.

[9] A. K. Das, "A secure and robust password-based remote user authentication scheme using smart cards for the integrated EPR information system," Journal of medical systems, vol. 39, no. 3, pp. 1–14, 2015.

[10] R. Madhusudhan and C. S. Nayak, "A robust authentication scheme for telecare medical information systems," Multimedia Tools and Applications, vol. 78, no. 11, pp. 15 255–15 273, 2019.

[11] P. Chandrakar and H. Om, "Cryptanalysis and extended three-factor remote user authentication scheme in multi-server environment," Arabian Journal for Science and Engineering, vol. 42, no. 2, pp. 765–786, 2017.

[12] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, "Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model," IEEE Transactions on Information Forensics and Security, vol. 12, no. 6, pp. 1382–1392, 2017.

[13] S. Garg, K. Kaur, G. Kaddoum, J. J. P. C. Rodrigues, and M. Guizani, "Secure and lightweight authentication scheme for smart metering infrastructure in smart grid," IEEE Transactions on Industrial Informatics, vol. 16, no. 5, pp. 3548–3557, 2020.

[14] S. A. Chaudhry, J. Nebhen, K. Yahya, and F. Al-Turjman, "A privacy enhanced authentication scheme for securing smart grid infrastructure," IEEE Transactions on Industrial Informatics, pp. 1–1, 2021.

[15] M. Tanveer, A. H. Zahid, M. Ahmad, A. Baz, and H. Alhakami, "LAKE-IoD: Lightweight authenticated key exchange protocol for the Internet of Drone environment," IEEE Access, vol. 8, pp. 155 645–155 659, 2020.

[16] M. Tanveer, G. Abbas, Z. H. Abbas, M. Waqas, F. Muhammad, and S. Kim, "S6AE: Securing 6LoWPAN using authenticated encryption scheme," Sensors, vol. 20, no. 9, p. 2707, 2020.

[17] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," Wireless Networks, vol. 25, no. 8, pp. 4737–4750, 2019.

[18] S. Qiu, G. Xu, H. Ahmad, and L. Wang, "A robust mutual authentication scheme based on elliptic curve cryptography for telecare medical information systems," IEEE Access, vol. 6, pp. 7452–7463, 2018.

[19] S. Son, J. Lee, M. Kim, S. Yu, A. K. Das, and Y. Park, "Design of secure authentication protocol for cloud-assisted telecare medical information system using blockchain," IEEE Access, vol. 8, pp. 192 177–192 191, 2020.

[20] L. Zhang, S. Zhu, and S. Tang, "Privacy protection for telecare medicine information systems using a chaotic map-based three-factor authenticated

key agreement scheme," IEEE Journal of Biomedical and health informatics, vol. 21, no. 2, pp. 465–475, 2016.

[21] D. He, S. Zeadally, N. Kumar, and W. Wu, "Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures," IEEE transactions on information forensics and security, vol. 11, no. 9, pp. 2052–2064, 2016.

[22] B. A. Alzahrani and A. Irshad, "A secure and efficient TMIS-based authentication scheme improved against zhang et al.'s scheme," Arabian Journal for Science and Engineering, vol. 43, no. 12, pp. 8239–8253, 2018.

[23] M. Tanveer, G. Abbas, and Z. H. Abbas, "LAS-6LE: A lightweight authentication scheme for 6LoWPAN environments," in 2020 14th International Conference on Open Source Systems and Technologies (ICOSST), 2020, pp. 1–6.

[24] L. Han, X. Tan, S. Wang, and X. Liang, "An efficient and secure three-factor based authenticated key exchange scheme using elliptic curve cryptosystems," Peer-to-peer Networking and Applications, vol. 11, no. 1, pp. 63–73, 2018.

[25] C. S. Nayak et al., "An improved user authentication scheme for electronic medical record systems." Multimedia Tools & Applications, vol. 79, 2020.

[26] S. A. Chaudhry, K. Yahya, M. Karuppiah, R. Kharel, A. K. Bashir, and Y. B. Zikria, "GCACS-IoD: A certificate based generic access control scheme for Internet of Drones," Computer Networks, vol. 191, p. 107999, 2021.

[27] O. Mir, T. van der Weide, and C.-C. Lee, "A secure user anonymity and authentication scheme using AVISPA for telecare medical information systems," Journal of medical systems, vol. 39, no. 9, pp. 1–16, 2015.

[28] R. Amin and G. Biswas, "An improved RSA based user authentication and session key agreement protocol usable in tmis," Journal of Medical Systems, vol. 39, no. 8, pp. 1–14, 2015.

[29] K. Renuka, S. Kumari, and X. Li, "Design of a secure three-factor authentication scheme for smart healthcare," Journal of medical systems, vol. 43, no. 5, pp. 1–12, 2019.

[30] D. Dharminder, D. Mishra, and X. Li, "Construction of RSA-based authentication scheme in authorized access to healthcare services," Journal of medical systems, vol. 44, no. 1, pp. 1–9, 2020.

[31] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," Journal of Network and Computer Applications, vol. 106, pp. 117–123, 2018.

[32] S. A. Chaudhry, H. Naqvi, and M. K. Khan, "An enhanced lightweight anonymous biometric based authentication scheme for tmis," Multimedia Tools and Applications, vol. 77, no. 5, pp. 5503–5524, 2018.

[33] A. Irshad, S. A. Chaudhry, S. Kumari, M. Usman, K. Mahmood, and M. S. Faisal, "An improved lightweight multiserver authentication scheme," International Journal of Communication Systems, vol. 30, no. 17, p. e3351, 2017.

[34] A. Irshad, S. A. Chaudhry, M. Shafiq, M. Usman, M. Asif, and A. Ghani, "A provable and secure mobile user authentication scheme for mobile cloud computing services," International Journal of Communication Systems, vol. 32, no. 14, p. e3980, 2019.

[35] J. Mo, Z. Hu, H. Chen, and W. Shen, "An efficient and provably secure anonymous user authentication and key agreement for mobile cloud computing," Wireless Communications and Mobile Computing, vol. 2019, 2019.

[36] S. Barman, A. K. Das, D. Samanta, S. Chattopadhyay, J. J. Rodrigues, and Y. Park, "Provably secure multi-server authentication protocol using fuzzy commitment," IEEE Access, vol. 6, pp. 38 578–38 594, 2018.

[37] R. Amin, T. Maitra, D. Giri, and P. Srivastava, "Cryptanalysis and improvement of an rsa based remote user authentication scheme using smart card," Wireless Personal Communications, vol. 96, no. 3, pp. 4629–4659, 2017.

[38] M. Nikooghadam, R. Jahantigh, and H. Arshad, "A lightweight authentication and key agreement protocol preserving user anonymity," Multimedia Tools and Applications, vol. 76, no. 11, pp. 13 401–13 423, 2017.

[39] R. Ali and A. K. Pal, "An efficient three factor–based authentication scheme in multiserver environment using ECC," International Journal of Communication Systems, vol. 31, no. 4, p. e3484, 2018.

[40] F. Wang, G. Xu, C. Wang, and J. Peng, "A provably secure biometrics-based authentication scheme for multiserver environment," Security and Communication Networks, vol. 2019, 2019.

[41] S. A. Chaudhry, "Combating identity de-synchronization: An improved lightweight symmetric key based authentication scheme for iov," Journal of Network Intelligence, vol. 6, pp. 656–667, 2021.

[42] M. Tanveer, H. Shah, S. A. Chaudhry, A. Naushad et al., "PASKE-IoD: Privacy-protecting authenticated key establishment for Internet of Drones," IEEE Access, vol. 9, pp. 145 683–145 698, 2021.

[43] M. Tanveer, N. Kumar, A. Naushad, S. A. Chaudhry et al., "A robust access control protocol for the smart grid systems," IEEE Internet of Things Journal, 2021.

[44] D. Dolev and A. Yao, "On the security of public key protocols," IEEE Transactions on information theory, vol. 29, no. 2, pp. 198–208, 1983.

[45] M. Tanveer, H. Shah, A. Alkhayyat, S. A. Chaudhry, M. Ahmad et al., "ARAP-SG: Anonymous and reliable authentication protocol for smart grids," IEEE Access, vol. 9, pp. 143 366–143 377, 2021.

[46] S. A. Chaudhry, "Correcting "palk: Password-based anonymous lightweight key agreement framework for smart grid"," International Journal of Electrical Power & Energy Systems, vol. 125, p. 106529, 2021.

[47] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2001, pp. 453–474.

[48] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, "Ascon," Submission to the CAESAR competition: http://ascon. iaik. tugraz. at, 2014.

[49] C. Beierle, A. Biryukov, L. C. dos Santos, J. Großschädl, L. Perrin, A. Udovenko, V. Velichkov, and Q. Wang, "Lightweight AEAD and hashing using the sparkle permutation family," IACR Transactions on Symmetric Cryptology, pp. 208–261, 2020.

[50] F. Abed, C. Forler, and S. Lucks, "General classification of the authenticated encryption schemes for the CAESAR competition," Computer Science Review, vol. 22, pp. 13–26, 2016.

[51] 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 of Things Journal, pp. 1–1, 2021.

[52] M. Tanveer, G. Abbas, Z. H. Abbas, M. Bilal, A. Mukherjee, and K. S. Kwak, "LAKE-6SH: Lightweight user authenticated key exchange for 6LoWPAN-based smart homes," IEEE Internet of Things Journal, pp. 1–1, 2021.

[53] C. J. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols," in International conference on computer aided verification. Springer, 2008, pp. 414–418.

[54] H. Arshad and M. Nikooghadam, "Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems," Journal of medical systems, vol. 38, no. 12, pp. 1–12, 2014.

[55] 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," Journal of medical systems, vol. 43, no. 1, pp. 1–22, 2019.