Anonymity preserving and Lightweight Multi-Medical Server Authentication Protocol for Telecare Medical Information System

Ruhul Amin, SK Hafizul Islam, Senior Member, IEEE, Prosanta Gope, Kim-Kwang Raymond Choo, Senior Member, IEEE, and Nachiket Tapas

***Abstract*—Electronic health systems, such as Telecare Medical Information System (TMIS), allow patients to exchange their health information with a medical center/doctor for diagnosis in real-time, and across borders. Given the sensitive nature of health information/medical data, ensuring the security of such systems is crucial. In this paper, we revisit Das et al.'s authentication protocol, which is designed to ensure patient anonymity and untraceability. Then, we demonstrate that the security claims are invalid, by showing how both security features can be compromised. We also demonstrate that the protocol suffers from new smartcard launch attacks. To mitigate such design flaws, we propose a new lightweight authentication protocol using the cryptographic hash function for TMIS. We then analyze the security of the proposed protocol using AVISPA and Scyther, two widely used formal specification tools. The performance analysis demonstrates that our protocol is more efficient than other competing protocols.**

***Index Terms*—AVISPA; Multi-Medical system; Authentication; Anonymity; Telecare Medical Information System; Scyther.**

I. Introduction

An efficient electronic health system can facilitate or inform medical decision-making based on accurate and up-to-date information, such as the most up-to-date health / medical condition of the patient. It can be challenging to make a medical diagnosis for a new patient since the medical practitioner does not access to the patient's prior medical history and other relevant information in real-time, since this is the first time that this medical practitioner is seeing the patient. This limitation can be mitigated in an electronic health system, since the medical practitioner from a participating healthcare institution can obtain real-time access to the patient's medical history and other relevant information stored in the server of another healthcare institution. Clearly, ensuring data confidentiality, integrity, and privacy is crucial in such systems. However, requirements may vary between different electronic health systems, for example due to the privacy legislation in the jurisdiction where the systems are hosted. Examples of electronic health systems include the following [1]:

- Electronic Health Record (ERH) system is designed to maintain patient records, and facilitate e-prescribing, clinical administration, e-booking and digital imaging, and archiving systems.
- Telemedicine and telehealth systems allow the dissemination and storage of health and other medical data over the Internet to facilitate medical diagnosis and other medical services; thus, improving patient access and optimizing use of digital health assets.
- Clinical health IT systems include radiology, nursing systems, computer-assisted diagnostics, medical imaging, and surgery training and planning systems. Such systems facilitate physicians in providing more accurate diagnoses and treatments to patients.
- Online health IT systems include online health portals, which can be accessed by patients to obtain information about health related services (e.g. book an appointment and check on the reports of one's recent laboratory tests).
- Specialized electronic health systems include those used by researchers for public health data collection and analysis, such as bio-statistical programs for infectious diseases, drug development, and outcomes analysis.

In a Telecare Medical Information System (TMIS), patient anonymity is crucial and violation of patient privacy can have serious legal ramifications. Hence, cryptographic techniques are generally used to protect patient anonymity and data privacy [4], [5], [6], [3], [7], [8], [9], [2], [14], [17]. In 2015, for example, Amin and Biswas [17] proposed a conceptual architecture for the distributed medical system, where any registered patient can directly exchange medical data with participating medical practitioner(s). The authors also proposed a hash function based authentication protocol using smartcard. They claimed that their protocol is efficient and secure against several security threats. Das et al. [18], however, revealed a number of security weaknesses in Amin and Biswas's protocol [17] and presented an improved protocol. Independently, Lin et al. [19] also noted that the protocol of Amin and Biswas is insecure against off-line password guessing attack,

R. Amin is with the Department of Computer Science and Engineering, Dr. Shyama Prasad Mukherjee International Institute of Information Technology, Naya Raipur, Chhattisgarh 493661, India. E-mail: amin_ruhul@live.com

S.H. Islam is with the Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India. Email:hafi786@gmail.com

P. Gope is with the School of Engineering and Computer Science, University of Hull Hull, HU6 7RX, United Kingdom. E-mail: prosanta@comp.nus.edu.sg

K-K. R. Choo is with the Department of Information Systems and Cyber Security and the Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249, USA. E-mail: raymond.choo@fulbrightmail.org

N. Tapas is with the Department of Engineering, University of Messina, Italy 98158. E-mail: ntapas@unime.it

insider attack and impersonation attack. In 2017, Liu et al. [20] also presented several security weaknesses of Das et al.'s [18] protocol as well as an improved protocol.

In the next section, we will revisit the protocol of Das et al. [18] before demonstrating in Section III that it is not capable of ensuring anonymity, as well as being insecure against traceability attack, and other smartcard related attacks. Hence, we propose an efficient authentication protocol that can be deployed in environment such as the TIMS architecture proposed by Amin and Biswas [17] – see Section IV. The security and performance analysis of our protocol are presented in Sections V and VI, respectively. We conclude the paper in Section VII.

II. Revisiting Das et al.'s Protocol

The protocol of Das et al. [18] consists of the following phases: (i) Medical-server registration phase, (ii) patient registration phase, (iii) login phase, (iv) authentication and session key agreement phase, (v) password and biometric update phase, and (vi) dynamic medical-server addition phase – see Sections II-A to II-D.

A. Medical-server Registration Phase

The medical-server MS_j $(1 \leq j \leq m)$ selects an identity ID_{mj}, and sends it to the central medical registration server MRS. Upon receiving the request, MRS calculates a secret key $X_{mj} = h(ID_{mj} \| X_c)$ for MS_j, where X_c denotes the secret key of MRS. Then, MRS sends X_{mj} to MS_j securely. Note that MS_j keeps $\langle ID_{mj}, X_{mj} \rangle$ in the database. For m' additional medical servers MS_j $(m'+1 \leq j \leq m+m')$, MRS selects an unique identity ID_{amj}, calculates $X_{amj} = h(ID_{amj} \| X_c)$, and stores $\langle ID_{amj}, X_c \rangle$ in the memory for all additional medical servers MS_j. These information will be used ensure scalability, if needed.

B. Patient Registration Phase

In this phase, the following operations/steps are undertaken.

Step 1. The patient U_p enters ID_p (identity), PW_p (password), and B_p (biometric) at the sensor device. U_p creates a random number K_p and computes $RPW_p = h(ID_p \| K_p \| PW_p)$. Further, U_p utilizes the fuzzy extractor [27], [28] to calculate $Gen(B_p) = (\sigma_p, \tau_u)$. Now, U_p sends $\langle ID_p, RPW_p \rangle$ to MRS via a secure network.

Step 2. On receiving $\langle ID_p, RPW_p \rangle$, MRS calculates $A_{mj} = h(ID_p \| X_{mj}) \oplus RPW_p$, $P_{mj} = h(ID_{mj} \| X_{mj}) \oplus RPW_p$ for all $(1 \leq j \leq m+m')$. Now, MRS issues a new smartcard and writes $\langle ID_{mj}, A_{mj}, P_{mj}, h(\cdot), Gen(\cdot), Rep(\cdot), t \rangle$ into the memory of the smartcard, where $1 \leq j \leq m+m'$ and t denotes error-tolerance threshold value for fuzzy extractor.

Step 3. On getting the smartcard, U_p computes $E_p = h(ID_p \| \sigma_p) \oplus K_p$ and $F_p = h(ID_p \| RPW_p \| \sigma_p)$. Finally, U_p inserts E_p and F_p into the smartcard.

C. Login Phase

This phase is executed to login to any of the participating medical servers, including those belonging to other healthcare institutions. All operations performed in this phase are given below.

Step 1. U_p inserts the smartcard into the smartcard-reader and enters ID_p, PW_p and B_p. Now, the smartcard calculates $\sigma_p^* = Rep(B_p, \tau_p)$ [27], [28], $K_p^* = h(ID_p \| \sigma_p^*) \oplus E_p$, $RPW_p^* = h(ID_p \| K_p^* \| PW_p)$, $F_p^* = h(ID_p \| RPW_p^* \| \sigma_p^*)$ and checks whether $F_p^* =?F_p$. If $F_p^* \neq F_p$, then the smartcard aborts the session, otherwise the inputted information is considered to be correct.

Step 2. The smartcard now creates a random number R_p and timestamp TS_p, then computes $MG_1 = A_{mj} \oplus RPW_p^* = h(ID_p \| X_{mj})$, $MG_2 = P_{mj} \oplus RPW_p^* = h(ID_{mj} \| X_{mj})$, $MG_3 = ID_p \oplus MG_2$, $MG_4 = ID_p \oplus MG_1 \oplus R_p$ and $MG_5 = h(MG_1 \| MG_3 \| MG_4 \| R_p \| TS_p)$. Now, the smartcard sends $\langle ID_{mj}, ID_{pk}, MG_3, MG_4, MG_5, TS_p \rangle$ as login message to MS_j via any unreliable network.

D. Authentication and Session Key Agreement Phase

This phase authenticates the involved entities and negotiates a common session key for secure information exchange.

Step 1. On receiving the login message $\langle ID_{mj}, ID_{pk}, MG_3, MG_4, MG_5, TS_p \rangle$ at time TS_{mj}, MS_j determines the timestamp's validity by verifying whether $|TS_{mj} - TS_p| \leq \triangle T$ holds, where $\triangle T$ denotes expected transmission delay. If the above condition is correct, then MS_j executes the next step, otherwise terminates the session.

Step 2. MS_j calculates $MG_6 = h(ID_{mj} \| X_{mj})$, $MG_7 = MG_3 \oplus MG_6 = ID_p$, $MG_8 = h(ID_p \| X_{mj})$, $MG_9 = MG_4 \oplus MG_7 \oplus MG_8 = R_p$, $MG_{10} = h(MG_8 \| MG_3 \| MG_4 \| MG_9 \| TS_p)$. MS_j now verifies whether $MG_{10} = ?MG_5$. If $MG_{10} \neq MG_5$, then MS_j aborts the session, otherwise authenticates the U_p. Note that, Das et al.'s protocol follows the same technique used in protocol [29] to mitigate replay and man-in-the-middle attacks.

Step 3. MS_j creates a random number R_{mj} and timestamp TS_{ms}, and then calculates $MG_{11} = h(ID_{mj} \| ID_{pk} \| X_{mk})$, $MG_{12} = ID_p \oplus MG_{11}$, $MG_{13} = h(ID_p \| X_{mk}) \oplus R_{mj}$, $MG_{14} = ID_p \oplus MG_9 \oplus R_{mj}$, $MG_{15} = h(ID_p \| MG_{11} \| MG_{12} \| MG_{13} \| MG_{14} \| MG_9 \| R_{mj} \| TS_{ms})$. MS_j sends a request message $\langle ID_{mj}, ID_{pk}, MG_{12}, MG_{13}, MG_{14}, MG_{15}, TS_{ms} \rangle$ to PS_k over any unreliable network.

Step 4. On receiving $\langle ID_{mj}$, ID_{pk}, MG_{12}, MG_{13}, MG_{14}, MG_{15}, $TS_{ms} \rangle$, PS_k first checks whether $|TS_{mk} - TS_{ms}| \leq \triangle T$ holds. If the condition is correct, then PS_k proceeds with further operation, otherwise terminates the current session. PS_k computes $MG_{16} = h(ID_{mj} \| ID_{mk} \| X_{mk})$, $MG_{17} = MG_{12} \oplus MG_{16} = ID_p$, $MG_{18} = MG_{13} \oplus h(MG_{17} \| X_{mk}) = R_{mj}$, $MG_{19} = MG_{14} \oplus MG_{17} \oplus MG_{18} = R_p$, $MG_{20} = h(MG_{17} \| MG_{16} \| MG_{12} \| MG_{13} \| MG_{14} \| MG_{19} \| MG_{18} \| TS_{ms})$. PS_k now checks whether $MG_{20} = MG_{15}$ holds. If $MG_{20} \neq MG_{15}$, then PS_k terminates the session, otherwise determines that both U_p and MS_j are authentic entities.

Step 5. PS_k creates random number R_{mk}, chooses the current timestamp TS_k, and computes $MG_{21} = h(MG_{17} \parallel X_{mk}) = h(ID_p \parallel X_{mk})$, $MG_{22} = MG_{17} \oplus MG_{19} \oplus R_{mk}$, $MG_{23} = MG_{21} \oplus R_{mk}$, $SK_{U_p,PS_k} = h(MG_{17} \parallel ID_{pk} \parallel MG_{19} \parallel R_{mk} \parallel MG_{21} \parallel TS_k)$, $MG_{24} = h(SK_{U_p,PS_k} \parallel MG_{22} \parallel MG_{23} \parallel MG_{19} \parallel R_{mk} \parallel TS_k)$. Now, PS_k sends $\langle ID_{mk}, MG_{22}, MG_{23}, MG_{24}, TS_k \rangle$ to U_p via any unreliable network.

Step 6. On receiving $\langle ID_{mk}, MG_{22}, MG_{23}, MG_{24}, TS_k \rangle$, the smartcard of U_p chooses a timestamp TS^*_p and checks whether the condition $|TS^*_p - TS_k| \leq \triangle T$ is correct. If it is incorrect, then the smartcard of U_p aborts the session, otherwise calculates $MG_{25} = MG_{22} \oplus (ID_p \oplus R_p) = R_{mk}$, $MG_{26} = MG_{23} \oplus MG_{25} = h(ID_p \parallel X_k)$, $SK^*_{U_p,PS_k} = h(ID_p \parallel ID_{mk} \parallel R_p \parallel MG_{25} \parallel MG_{26} \parallel TS_k)$, $MG_{27} = h(SK^*_{U_p,PS_k} \parallel MG_{22} \parallel MG_{23} \parallel R_p \parallel MG_{25} \parallel TS_k)$. Now, the smartcard of U_p checks whether $MG_{27} =?MG_{24}$ holds. If $MG_{27} \neq MG_{24}$, then the smartcard of U_p terminates the session, otherwise, believes that PS_k is authentic entity and the common session session key is $SK^*_{U_p,PS_k} = SK_{U_p,PS_k}$.

III. Security Weaknesses in Das et al.'s Protocol

We will now reveal the insecurity of Das et al.'s protocol.

A. Loss of Patient Anonymity

As previously mentioned, the anonymity of a patient is highly important [17]. Das et al.'s protocol claimed that the patients are anonymous in their system. However, we will now demonstrate how this property can be violated.

1) Loss of Anonymity using Smartcard Information: In this method, we assume that a legitimate patient can act as a malicious patient or adversary, who has the smartcard of a legitimate patient, and thus can retrieve smartcard information $\langle ID_{mj}, A_{mj}, P_{mj}, E_p, F_p, h(\cdot), Gen(\cdot), Rep(\cdot), t \rangle$ (e.g. using the technique proposed in [30], [31]). As the malicious patient is legitimate, he/she can compute $h(ID_{mj} \parallel X_{mj})$ as $(P_{mj} \oplus PWR_p)$ using his/her PWR_p. Since the login information MG_3 is transmitted through an unreliable network, the malicious patient can intercept MG_3. Now, the malicious patient can find ID_p as $MG_3 \oplus h(ID_{mj} \parallel X_{mj})$. Thus, Das et al.'s protocol does not ensure patient anonymity.

2) Loss of Anonymity using Login and Authentication Messages: Another legitimate patient may seek to leak medical server information and become a malicious patient (e.g. registering with the server in order to exploit this vulnerability), which we denote as $\mathcal{A}$. In Das et al.'s protocol, the system's information MG_{11} is static and can easily be computed by the legitimate patient as $ID_p \oplus MG_{12}$. Note that it is not difficult to make the information MG_{11} public. The malicious user now can compute ID_p as $MG_{11} \oplus MG_{12}$ of any legitimate patient only using the login and authentication messages, which are transmitted over an insecure network. In other words, anyone can break the anonymity assurance in Das et al.'s protocol.

B. Loss of Traceability Attack

For a stronger sense of anonymity, traceability is always regarded as an important property. Now, we demonstrate that Das et al.'s protocol cannot ensure the traceability property. We assume that the adversary $\mathcal{A}$ intercepts two successive login messages $\langle ID_{mj}, ID_{pk}, MG_3, MG_4, MG_5, TS_p \rangle$ and $\langle ID'_{mj}, ID'_{pk}, MG'_3, MG'_4, MG'_5, TS'_p \rangle$, where $MG_2 = P_{mj} \oplus RPW^*_p = h(ID_{mj} \parallel X_{mj})$, $MG_3 = ID_p \oplus MG_2$. Note that the computation of MG_3 depends only on the patient's and server's information. If $(MG_3 =?MG'_3)$, then $\mathcal{A}$ believes that the two login messages are from the same patient. Thus, Das et al.'s protocol is not sure against traceability attack.

C. New Smartcard Launch Attack

This is another common security feature typically required of smartcards, where an adversary $\mathcal{A}$ attempts to issue new smartcard by using a legitimate patient's identity. In order to launch this attack, the following operations are required to be performed.

Step 1. In Section III-A2, we present the approach to find ID_p of U_p. Therefore, $\mathcal{A}$ performs the same approach to know the ID_p of U_p

Step 2. Now, $\mathcal{A}$ computes $MG_{11} = h(ID_{mj} \parallel X_{mj}) = MG_{12} \oplus ID_p$.

Step 3. $\mathcal{A}$ chooses a new password PW^n_p, biometric B^n_p and computes $RPW^n_p = h(ID_p \parallel K^n_p \parallel PW^n_p)$, $A^n_{mj} = A_{mj} \oplus RPW^*_p \oplus RPW^n_p$, $P^n_{mj} = P_{mj} \oplus RPW^*_p \oplus RPW^n_p$, $Gen(B^n_p) = (\sigma^n_p, \tau^n_p)$, $E^n_p = h(ID_p \parallel \sigma^n_p) \oplus K^n_p$, $F^n_p = h(ID_p \parallel RPW^n_p \parallel \sigma^n_p)$, where RPW^*_p can be computed as discussed in Section III-A1. Finally, $\mathcal{A}$ stores $\langle ID_{mj}, A^n_{mj}, P^n_{mj}, E^n_p, F^n_p, h(\cdot), Gen(\cdot), Rep(\cdot), t \rangle$ into the new smartcard, where $1 \leq j \leq m + m'$. Thus, $\mathcal{A}$ can launch the above attack successfully.

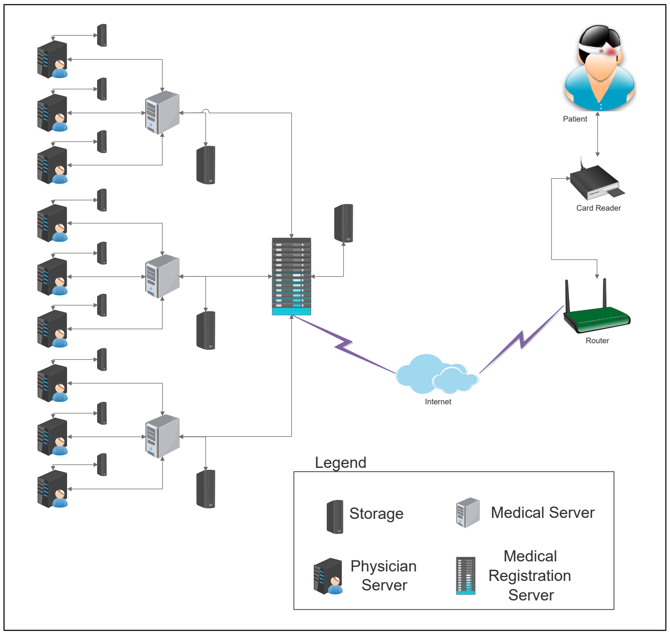


Fig. 1. Multi-medical system framework [17]

IV. Proposed Protocol

Fig. 1 outlines the medical system architecture proposed in [17], which comprises medical registration server, medical server, physician server and patients. The medical registration server is mainly responsible for providing registration service to all medical servers. In this architecture, there are several medical servers located in different locations for providing registration services to the physician servers and the patients. Note that the physician servers may be under the control of any medical server. In addition, it participates in the negotiation of session key between a physician server and patient. The patients may seek medical treatment from anywhere and anytime. After performing the registration with a medical server, the patient logins into the system to negotiate a session key with a physician server. This session key will be used to exchange some medical information of the patient.

Our proposed protocol consists of nine phase, namely: setup, medical-server registration, physician-server registration, patient registration, login, authentication and session key negotiation, new physician-server addition, password renewal, and biometric renewal. Table I summarizes the notations used in the protocol.

TABLE I
Notations

Symbol	Description
MRS	Central-medical-registration-server
U_p	Patient, $(1 \leq p \leq m)$
MS_j	Medical-server, $(1 \leq j \leq n)$
PS_k	Physician-server, $(1 \leq k \leq l)$
ID_{mj}	Identity of the MS_j
ID_{pk}	Identity of the PS_k
ID_p	Identity of the U_p
TID_p	Temporary unique identity of the U_p
B_p	Biometric of the U_p
PW_p	Password of the U_p
MK_{shrd}	Common key between all the MS_j
SK	Session key of the protocol
$H(\cdot)$	Bio-hashing function
$h(\cdot)$	Hash function $\{0,1\}^* \rightarrow \{0,1\}^k$
$\parallel$	Concatenation operation $(a \parallel b)$
$\oplus$	Bitwise XOR operation $(a \oplus b)$

A. Setup Phase

MRS (medical registration server) selects a master key MK_{shrd} and shares it among all medical servers MS_j $(1 \leq j \leq n)$ securely. Note that, this key cannot be compromised under any circumstances.

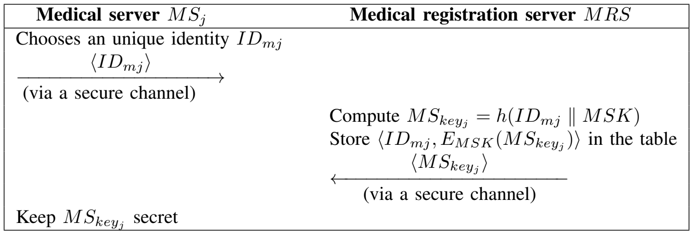

Medical server MS_j
Medical registration server MRS
Chooses an unique identity ID_{mj}
$\langle ID_{mj} \rangle$
(via a secure channel)
Compute $MS_{key_j} = h(ID_{mj} \parallel MSK)$
Store $\langle ID_{mj}, E_{MSK}(MS_{key_j}) \rangle$ in the table
$\langle MS_{key_j} \rangle$
(via a secure channel)
Keep MS_{key_j} secret

Fig. 2. Medical server registration phase of our protocol.

B. Medical Server Registration Phase

Each medical server MS_j $(1 \leq j \leq n)$ needs to register with MRS before accessing any service. MS_j chooses an unique identity ID_{mj} and sends it to MRS via a secure channel. On receiving ID_{mj}, MRS calculates $MS_{key_j} = h(ID_{mj} \parallel MSK)$ as a secret key of MS_j and stores $\langle ID_{mj}, E_{MSK}(MS_{key_j}) \rangle$ in the table for all MS_j $(1 \leq j \leq n)$. Finally, MRS sends $\langle MS_{key_j} \rangle$ to MS_j via a secure channel. Note that, MS_j keeps MS_{key_j} secret and none can obtain it by any means. Fig. 2 also describes this phase.

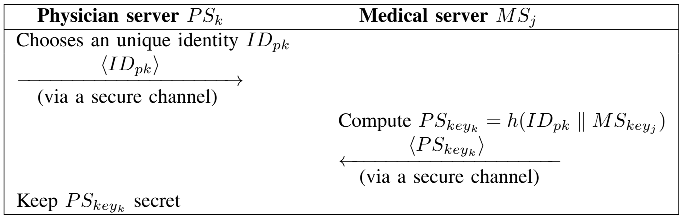

Physician server PS_k
Medical server MS_j
Chooses an unique identity ID_{pk}
$\langle ID_{pk} \rangle$
(via a secure channel)
Compute $PS_{key_k} = h(ID_{pk} \parallel MS_{key_j})$
$\langle PS_{key_k} \rangle$
(via a secure channel)
Keep PS_{key_k} secret

Fig. 3. Physician server registration phase of our protocol.

C. Physician Server Registration Phase

The main objective of this phase (see Fig. 3) is to negotiate a session key between the patient U_p and a physician server PS_k, which is managed by a doctor. Therefore, all physician servers PS_k $(1 \leq k \leq l)$ must register with the home medical server MS_j. Specifically, PS_k chooses a unique identity ID_{pk} and sends it to MS_j via a secure channel. On getting ID_{pk}, MS_j calculates $PS_{key_k} = h(ID_{pk} \parallel MS_{key_j})$ and sends to PS_k securely.

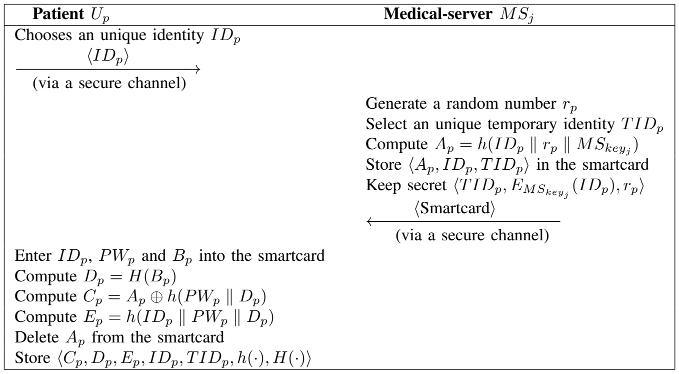

Patient U_p
Medical-server MS_j
Chooses an unique identity ID_p
$\langle ID_p \rangle$
(via a secure channel)
Generate a random number r_p
Select an unique temporary identity TID_p
Compute $A_p = h(ID_p \parallel r_p \parallel MS_{key_j})$
Store $\langle A_p, ID_p, TID_p \rangle$ in the smartcard
Keep secret $\langle TID_p, E_{MS_{key_j}}(ID_p), r_p \rangle$
⟨Smartcard⟩
(via a secure channel)
Enter ID_p, PW_p and B_p into the smartcard
Compute $D_p = H(B_p)$
Compute $C_p = A_p \oplus h(PW_p \parallel D_p)$
Compute $E_p = h(ID_p \parallel PW_p \parallel D_p)$
Delete A_p from the smartcard
Store $\langle C_p, D_p, E_p, ID_p, TID_p, h(\cdot), H(\cdot) \rangle$

Fig. 4. Patient registration phase of our protocol.

D. Patient Registration Phase

In this phase (see Fig. 4), a patient U_p $(1 \leq p \leq m)$ needs to be registered with a nearby medical server MS_j in order to access physician servers located in different places.

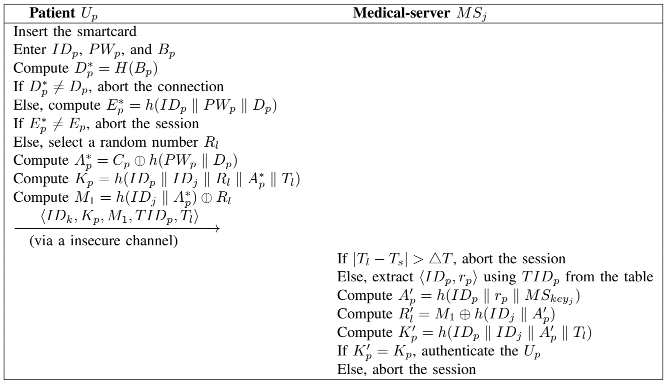
Step 1. U_p chooses a unique identity ID_p and sends it to MS_j securely.

Step 2. MS_j computes $A_p = h(ID_p \parallel r_p \parallel MS_{key_j})$ and stores $\langle A_p, ID_p, TID_p \rangle$ it in the memory of a smartcard, where r_p is the random number selected by MS_j and TID_p is the unique temporary identity for U_p. Then, MS_j keeps a secret table, which includes a tuple of the form

$\langle TID_p, E_{MS_{key_j}}(ID_p), r_p\rangle$ for each $U_p (1 \leq p \leq m)$. Finally, MS_j delivers the smartcard to the U_p through an off-line mode.

Step 3. On receiving the smartcard, U_p enters his/her identity ID_p, password PW_p, and biometric B_p into the smartcard. Now, the smartcard calculates $D_p = H(B_p)$, $C_p = A_p \oplus h(PW_p \parallel D_p)$, and $E_p = h(ID_p \parallel PW_p \parallel D_p)$, where $H(\cdot)$ is the bio-hashing function [8].

Step 4. U_p deletes A_p from the smartcard. Finally, the smartcard holds $\langle C_p, D_p, E_p, ID_p, TID_p, h(\cdot), H(\cdot)\rangle$. The



Patient U_p	**Medical-server** MS_j
Insert the smartcard	
Enter ID_p, PW_p, and B_p	
Compute $D_p^* = H(B_p)$	
If $D_p^* \neq D_p$, abort the connection	
Else, compute $E_p^* = h(ID_p \parallel PW_p \parallel D_p)$	
If $E_p^* \neq E_p$, abort the session	
Else, select a random number R_l	
Compute $A_p^* = C_p \oplus h(PW_p \parallel D_p)$	
Compute $K_p = h(ID_p \parallel ID_j \parallel R_l \parallel A_p^* \parallel T_l)$	
Compute $M_1 = h(ID_j \parallel A_p^*) \oplus R_l$	
$\xrightarrow{\langle ID_k, K_p, M_1, TID_p, T_l\rangle}$	
(via a insecure channel)	
	If $\lvert T_l - T_s\rvert > \triangle T$, abort the session
	Else, extract $\langle ID_p, r_p\rangle$ using TID_p from the table
	Compute $A_p' = h(ID_p \parallel r_p \parallel MS_{key_j})$
	Compute $R_l' = M_1 \oplus h(ID_j \parallel A_p')$
	Compute $K_p' = h(ID_p \parallel ID_j \parallel A_p' \parallel T_l)$
	If $K_p' = K_p$, authenticate the U_p
	Else, abort the session

Fig. 5. Login and authentication phase of our protocol.

E. Login and Authentication Phase

This phase (Fig. 5) is invoked whenever a patient needs to access medical data from a physician server. To complete this phase, the smartcard generates login message on inputting patient information and then forwards the message to MS_j, where he/she has executed the registration phase.

Step 1. At the beginning, U_p inserts the smartcard into the terminal, and enters ID_p, PW_p, and B_p. Then, the smartcard computes $D_p^* = H(B_p)$. If $D_p^* \neq D_p$, then the smartcard of U_p aborts the session, otherwise calculates $E_p^* = h(ID_p \parallel PW_p \parallel D_p)$. If $E_p^* \neq E_p$, then the smartcard of the U_p aborts the session, otherwise, executes further operations.

Step 2. The smartcard of U_p selects a random number R_l, and timestamp T_l. The smartcard of U_p computes $A_p^* = C_p \oplus h(PW_p \parallel D_p)$, $K_p = h(ID_p \parallel ID_j \parallel R_l \parallel A_p^* \parallel T_l)$, $M_1 = h(ID_j \parallel A_p^*) \oplus R_l$.

Step 3. Now, the smartcard of U_p provides a option to choose the identity ID_k of the physician server PS_k, from which the patient wishes to access the service. The smartcard then sends $\langle ID_k, K_p, M_1, TID_p, T_l\rangle$ to MS_j through an unreliable network.

Step 4. Whenever MS_j receives the login message $\langle ID_k, K_p, M_1, TID_p, T_l\rangle$, it checks whether $|T_l - T_s| \leq \triangle T$ holds, where T_s is current timestamp. If it holds, then MS_j extracts $\langle ID_p, r_p\rangle$ from the table using TID_p and calculates $A_p' = h(ID_p \parallel r_p \parallel MS_{key_j})$, $R_l' = M_1 \oplus h(ID_j \parallel A_p')$, $K_p' = h(ID_p \parallel ID_j \parallel A_p' \parallel T_l)$. If $K_p' \neq K_p$, then MS_j aborts the session, otherwise, authenticates the U_p. The

F. Session Key Agreement Phase

This phase (Fig. 6) is used to execute the mutual authentication and session between the entities in the protocol execution.

Step 1. After the execution of the login phase, MS_j receives the login message with ID_k as the parameter. In our protocol, the execution of Cases 1 and 2 depends on the existence of ID_p. If ID_k is present in the local database of MS_j, then Case 1 will be executed to allow the patient to negotiate a session key between U_p and PS_k. Here, ID_k is the identity of PS_k. On the other hand, if ID_k is not found in the local database, then MS_j searches in the global database. If MS_j is found, then we have Case 2.

Case 1.

Step 1-1. MS_j calculates $PS_{key_k} = h(ID_k \parallel MS_{key_j})$, $L_p = h(ID_k \parallel TID_p \parallel PS_{key_k}) \oplus R_l'$, $N_p = h(TID_p \parallel PS_{key_k}) \oplus ID_p$, $O_p = h(ID_p \parallel ID_j \parallel ID_k \parallel R_l' \parallel T_s)$ and then sends $\langle L_p,\ N_p,\ O_p,\ ID_j, T_s,\ TID_p\rangle$ to the PS_k over an unreliable network.

Step 1-2. On receiving $\langle L_p,\ N_p,\ O_p,\ ID_j, T_s,\ TID_p\rangle$, PS_k first checks whether $|T_s - T_k| \leq \triangle T$ holds, where T_k is the current timestamp. If the above verification fails, then PS_k terminates the connection, otherwise computes $R_l'' = L_p \oplus h(ID_k \parallel TID_p \parallel PS_{key_k})$, $ID_p' = N_p \oplus h(TID_p \parallel PS_{key_k})$, and $O_p'' = h(ID_p' \parallel ID_j \parallel ID_k \parallel R_l'' \parallel T_s)$. If $O_p'' \neq O_p$, then PS_k aborts the connection, otherwise executes further computations.

Step 1-3. PS_k now selects a random number R_p, a session key SK randomly, and then calculates $P_p = h(ID_p \parallel SK \parallel ID_k \parallel T_k \parallel h(ID_p \parallel PS_{key_k}))$, $Q_p = R_l'' \oplus h(ID_p \parallel PS_{key_k})$, $S_p = SK \oplus h(ID_p \parallel R_l'' \parallel T_k)$. Finally, PS_k sends $\langle P_p, Q_p, S_p, T_k\rangle$ to U_p via an unreliable network.

Step 1-4. U_p first takes whether $|T_t - T_k| \leq \triangle T$ holds, where T_t is the current timestamp. If the above condition is correct, then U_p extracts $h(ID_p \parallel PS_{key_k})$ using R_l and SK' and $\langle ID_p, R_l\rangle$ from Q_p and S_p, respectively. Now, U_p further calculates $P_p' = h(ID_p \parallel SK' \parallel ID_k \parallel T_k \parallel h(ID_p \parallel PS_{key_k}))$. If $P_p' = P_p$, then U_p authenticates the PS_k, otherwise terminates the session.

Step 1-5. In order to resist traceability attack, U_p sends an acknowledgment to MS_j that the session is correctly executed. On receiving the acknowledgment, MS_j generates a random unique identity $TID_p' \neq TID_p$ and computes $M_2 = TID_p' \oplus R_l$, $M_3 = h(ID_p \parallel TID_p' \parallel R_l)$, and sends $\langle M_2, M_3\rangle$ to U_p over an unreliable network.

Step 1-6. On receiving $\langle M_2, M_3\rangle$, U_p extracts TID_p' as $TID_p' = M_2 \oplus R_l$ and $M_3' = h(ID_p \parallel TID_p' \parallel R_l)$. If the condition $M_3' = M_3$ is correct, then the smartcard of U_p replaces TID_p with TID_p' in the memory.

Case 2.

In this case, MS_j extracts the identity ID_{mj} of MS_j, which is managed by the the physician server PS_k with identity ID_{pk}. MS_j executes the following steps to negotiate a secure session key between U_j and PS_k.

Step 2-1. The home medical server MS_j computes $Y_1 = ID_p \oplus h(ID_{mj}' \parallel T_1 \parallel MK_{shrd})$ and $Y_2 = h(ID_{mj} \parallel ID_{mj}' \parallel MK_{shrd} \parallel T_1)$, where ID_{mj}' is the identity of the foreign medical server MS_j. Finally, MS_j sends $\langle Y_1, Y_2, T_1, ID_{mj}\rangle$ through an insecure network to MS_j, which is under the control of PS_k.

Step 2-2. On receiving $\langle Y_1, Y_2, T_1, ID_j\rangle$, MS_j first checks whether $|T_1 - T_2| \leq \triangle T$ holds. If the verification fails, then

Medical server MS_j	**Physician-server** PS_k	**Patient** U_p
MS_j checks whether ID_k is in the local database If ID_k is found, execute Case 1 Else, executes Case 2 **Case 1:** Compute $PS_{key_k} = h(ID_{pk} \parallel MS_{key_j})$ Compute $L_p = h(ID_{pk} \parallel TID_p \parallel PS_{key_k}) \oplus R'_l$ Compute $N_p = h(TID_p \parallel PS_{key_k}) \oplus ID_p$ Compute $O_p = h(ID_p \parallel ID_{mj} \parallel ID_{pk} \parallel R'_l \parallel T_s)$ $\xrightarrow{\langle L_p, N_p, O_p, ID_{mj}, T_s, TID_p \rangle}$ (via an insecure channel)		
	If $\|T_s - T_k\| > \triangle T$, abort the session Else, compute $R''_l = L_p \oplus h(ID_k \parallel TID_p \parallel PS_{key_k})$ Compute $ID'_p = N_p \oplus h(TID_p \parallel PS_{key_k})$ Compute $O''_p = h(ID'_p \parallel ID_{mj} \parallel ID_{pk} \parallel R''_l \parallel T_s)$ If $O''_p \neq O_p$, abort the session Else, generate a random number R_p and a session key SK Compute $P_p = h(ID_p \parallel SK \parallel ID_{pk} \parallel T_k \parallel h(ID_p \parallel PS_{key_k}))$ Compute $Q_p = R''_l \oplus h(ID_p \parallel PS_{key_k})$ Compute $S_p = SK \oplus h(ID_p \parallel R''_l \parallel T_k)$ $\xrightarrow{\langle P_p, Q_p, S_p, T_k \rangle}$ (via an insecure channel)	
		If $\|T_t - T_k\| > \triangle T$, abort the session Else, extract $h(ID_p \parallel PS_{key_k})$ using R_l Extract SK' using $\langle ID_p, R_l \rangle$ Compute $P'_p = h(ID_p \parallel SK' \parallel ID_{pk} \parallel T_k \parallel h(ID_p \parallel PS_{key_k}))$ If $P'_p \neq P_p$, abort the session Else, accept PS_k $\xleftarrow{\langle \text{Acknowledgement} \rangle}$ (via an insecure channel)
Generate $TID'_p \neq TID_p$ Compute $M_2 = TID'_p \oplus R_l$ Compute $M_3 = h(ID_p \parallel TID'_p \parallel R_l)$ $\xrightarrow{\langle M_2, M_3 \rangle}$ (via an insecure channel)		
		Compute $TID'_p = M_2 \oplus R_l$ Compute $M'_3 = h(ID_p \parallel TID'_p \parallel R_l)$ If $M'_3 \neq M_3$, abort the session Else, replace TID_p with TID'_p in the smartcard

Fig. 6. Session key agreement phase of our protocol (Case 1).

Home medical server MS_j	**Foreign medical server** MS_j	**Physician server** PS_k	**Patient** U_p
Assume that ID_{pk} is found in the global database Compute $Y_1 = ID_p \oplus h(ID'_{mj} \parallel T_1 \parallel MK_{shrd})$ Compute $Y_2 = h(ID_{mj} \parallel ID'_{mj} \parallel MK_{shrd} \parallel T_1)$ $\xrightarrow{\langle Y_1, Y_2, T_1, ID_{mj} \rangle}$ (via a insecure channel)			
	If $\|T_1 - T_2\| > \triangle T$, aborts the session Else, compute $Y'_2 = h(ID_{mj} \parallel ID'_{mj} \parallel MK_{shrd} \parallel T_1)$ If $Y'_2 \neq Y_2$, aborts the session Else, extract $ID_p = Y_1 \oplus h(ID'_{mj} \parallel T_1 \parallel MK_{shrd})$ Compute $Y_3 = h(ID_{pk} \parallel ID_p \parallel T_1 \parallel PS_{key_j})$ Compute $Y_4 = ID_p \oplus h(ID_{pk} \parallel PS_{key_j} \parallel T_1)$ $\xrightarrow{\langle Y_3, Y_4, T_2 \rangle}$ (via a insecure channel)		
		If $\|T_2 - T_3\| > \triangle T$ aborts the session Else, generate a random session key SK Compute $ID'_p = Y_4 \oplus h(ID_{pk} \parallel PS_{key_j} \parallel T_1)$ Compute $Y'_3 = h(ID_{pk} \parallel ID'_p \parallel T_1 \parallel PS_{key_j})$ If $(Y'_3 \neq Y_3)$, abort the session Compute $Y_5 = ID'_p \oplus SK$ Compute $Y_6 = h(ID'_p \parallel SK \parallel ID_{pk} \parallel T_3)$ $\xrightarrow{\langle Y_5, y_6, T_3 \rangle}$ (via a insecure channel)	
			If $\|T_3 - T_4\| > \triangle T$, aborts the session Else, compute $Y'_6 = h(ID_p \parallel SK \parallel ID_{pk} \parallel T_3)$ If $Y'_6 \neq Y_6$, the U_p abort the session $\xleftarrow{\langle \text{Acknowledgement} \rangle}$ (via a insecure channel)
Receives acknowledgment Generates TID'_p where $TID'_p \neq TID_p$ Computes $M_{22} = TID'_p \oplus R_l$ Computes $M_{33} = h(ID_p \parallel TID'_p \parallel R_l)$ $\xrightarrow{\langle M_{22}, M_{33} \rangle}$ (via a insecure channel)			
			Receives messages from MS_j Computes $TID'_p = M_{22} \oplus R_l$ Computes $M'_{33} = h(ID_p \parallel TID'_p \parallel R_l)$ If $\|M'_{33} \neq M_{33}\|$, aborts the session replaces TID_p with TID'_p in the smartcard

Fig. 7. Session key agreement phase of our protocol (Case 2).

MS_j aborts the session, otherwise, computes $Y'_2 = h(ID_{mj} \parallel ID'_{mj} \parallel MK_{shrd} \parallel T_1)$. If $(Y'_2 \neq Y_2)$, then MS_j aborts the connection, otherwise extracts $ID_p = Y_1 \oplus h(ID'_{mj} \parallel T_1 \parallel MK_{shrd})$, and computes $Y_3 = h(ID_{pk} \parallel ID_p \parallel T_1 \parallel$

$PS_{key_j})$, $Y_4 = ID_p \oplus h(ID_{pk} \parallel PS_{key_j} \parallel T_1)$. MS_j sends $\langle Y_3, Y_4, T_2 \rangle$ to the PS_k through an insecure network.

Step 2-3. On receiving $\langle Y_3, Y_4, T_2 \rangle$, PS_k first checks whether $|T_2 - T_3| \leq \triangle T$ holds, where T_3 is the current timestamp. If the timestamp verification fails, then PS_k aborts the session, otherwise, generates a random session key SK and computes $ID'_p = Y_4 \oplus h(ID_{pk} \parallel PS_{key_j} \parallel T_1)$, $Y'_3 = h(ID_k \parallel ID'_p \parallel T_1 \parallel PS_{key_j})$. If $Y'_3 \neq Y_3$, then PS_k aborts the connection, otherwise, computes $Y_5 = ID'_p \oplus SK$, $Y_6 = h(ID'_p \parallel SK \parallel ID_{pk} \parallel T_3)$ and sends $\langle Y_5, Y_6, T_3 \rangle$ to U_p through an insecure network.

Step 2-4. U_p checks whether $|T_3 - T_4| \leq \triangle T$ holds, where T_4 is the current timestamp. If the timestamp verification fails, then U_p aborts the session, otherwise, computes $Y'_6 = h(ID_p \parallel SK \parallel ID_{pk} \parallel T_3)$. If $Y'_6 \neq Y_6$, then U_p aborts the connection, otherwise, sends an acknowledgment to the home medical server MS_j to confirm that the session key is negotiated with the correct PS_k.

Step 2-5. On receiving the acknowledgment, MS_j generates a new random unique identity $TID'_p \neq TID_p$ and computes $M_{22} = TID'_p \oplus R_l$, $M_{33} = h(ID_p \parallel TID'_p \parallel R_l)$, and sends $\langle M_{22}, M_{33} \rangle$ to U_p over an unreliable network.

Step 2-6. On receiving $\langle M_{22}, M_{33} \rangle$, U_p extracts TID'_p as $TID'_p = M_{22} \oplus R_l$ and $M'_{33} = h(ID_p \parallel TID'_p \parallel R_l)$. If the condition $M'_{33} = M_{33}$ is correct, then the smartcard of the U_p replaces TID_p with TID'_p in the memory.

Fig. 7 also outlines this phase.

G. New Physician Server Addition Phase

It is also necessary to add new physician server(s) to the system for scalability and achieve higher system reliability. Thus, our protocol supports the addition of new physician server. The new physician server PS_t first chooses a unique identity ID_{pt} and then sends it to MS_j securely. Then, MS_j performs $PS_{key_t} = h(ID_{pt} \parallel MS_{key_j})$ and sends it to PS_t through a secure network. Accordingly, our protocol can increase the flexibility of the system by adding new physician server.

H. Password Update Phase

The password PW_p of patient U_p can be updated (periodically).

Step 1. U_p inserts the smartcard to the terminal and enters ID_p, PW_p, and B_p. Then, the smartcard computes $D^*_p = H(B_p)$. If $D^*_p \neq D_p$, then the smartcard aborts the session, otherwise, computes $E^*_p = h(ID_p \parallel PW_p \parallel D_p)$. If $E^*_p \neq E_p$, then the smartcard aborts the session, otherwise requests to enter a new password.

Step 2. Suppose that U_p inputs a new password PW^{new}_p. The smartcard now performs $A_p = C_p \oplus h(PW_p \parallel D_p)$, $C^{new}_p = A_p \oplus h(PW^{new}_p \parallel D_p)$, $E^{new}_p = h(ID_p \parallel PW^{new}_p \parallel D_p)$ and finally includes $\langle C^{new}_p, E^{new}_p \rangle$ into the memory and drops $\langle C_p, E_p \rangle$.

I. Biometric Renewal Phase

This phase updates the biometric information of the registered patient. First, executes Step 1 in Section IV-H to verify the correctness of ID_p, PW_p, and B_p. If all of these are correct, then the smartcard requests to enter new biometric information. Suppose that U_p enters B^{new}_p as the new biometric information. Now the smartcard computes $D^{new}_p = H(B^{new}_p)$, $C^{new}_p = A_p \oplus h(PW_p \parallel D^{new}_p)$, and $E^{new}_p = h(ID_p \parallel PW^{new}_p \parallel D^{new}_p)$ and includes $\langle D^{new}_p, C^{new}_p, E^{new}_p \rangle$ by replacing the old information $\langle D_p, C_p, E_p \rangle$.

V. Security Analysis

This section evaluates the security of the proposed protocol.

A. Capabilities of the Adversary

In this section, we analyze that whether an adversary can launch attacks or not based on our protocol information under the assumption mentioned in [21]. At first, our objective is to examine whether the adversary could calculate private information or not from the public information. The following cases will demonstrate the above objectives.

- **Case 1:** We assume that an adversary $\mathcal{A}$ can get the smartcard of the U_p and easily extract confidential information to break the security of the proposed protocol by using the processes described in [30], [31]. In our protocol, the smartcard includes $\langle C_p, D_p, E_p, ID_j, TID_p, h(\cdot), H(\cdot) \rangle$, where $D_p = H(B_p)$, $C_p = A_p \oplus h(PW_p \parallel D_p)$, and $E_p = h(ID_p \parallel PW_p \parallel D_p)$, $A_p = h(ID_p \parallel r_p \parallel MS_{key_j})$. The objective of the $\mathcal{A}$ is to calculate the private information $\langle ID_p, PW_p, A_p \rangle$ using the information obtained from the smartcard. Though $\mathcal{A}$ knows D_p, it is infeasible to calculate ID_p, PW_p from the known information E_p owing to the difficulties of the inversion of the hash function $h(\cdot)$. In the similar fashion, the $\mathcal{A}$ is not able to find A_p from C_p. Therefore, our proposed system is still secure even if the smartcard of the U_p is available to the $\mathcal{A}$.
- **Case 2.** In our protocol, the smartcard sends a login information $\langle ID_{pk}, K_p, M_1, TID_p, T_l \rangle$ to the MS_j through a public network. If $\mathcal{A}$ monitors the network, he/she can easily intercept all these information, where $A^*_p = C_p \oplus h(PW_p \parallel D_p)$, $K_p = h(ID_p \parallel ID_{mj} \parallel R_l \parallel A^*_p \parallel T_l)$, $M_1 = h(ID_{mj} \parallel A^*_p) \oplus R_l$. In this case, the confidential information are $\langle A_p, ID_p, PW_p, R_l \rangle$. In our protocol, we use a one-way hash function $h(\cdot)$ to compute K_p and M_1. All the information $\langle A_p, ID_p, PW_p, R_l \rangle$ are protected due to the $h(\cdot)$, and therefore the $\mathcal{A}$ cannot find these confidential information. Thus, if the $\mathcal{A}$ obtains the message $\langle ID_k, K_p, M_1, TID_p, T_l \rangle$ from the public network, he/she cannot break the security of the proposed system.
- **Case 3.** In our protocol, the MS_j sends the message $\langle L_p, N_p, O_p, ID_{mj}, T_s, TID_p \rangle$ to the PS_k through an unreliable network. Therefore, the $\mathcal{A}$ can capture all these information, where $PS_{key_k} = h(ID_{pk} \parallel MS_{key_j})$, $L_p = h(ID_{pk} \parallel TID_p \parallel PS_{key_k}) \oplus R'_l$, $N_p = h(TID_p \parallel PS_{key_k}) \oplus ID_p$, $O_p = h(ID_p \parallel ID_{mj} \parallel ID_{pk} \parallel R'_l \parallel T_s)$. The objective of $\mathcal{A}$ is to find $\langle PS_{key_k}, R_l, ID_p \rangle$. Though the $\mathcal{A}$ knows ID_{pk}, however, he/she is not able to compute PS_{key_k} due to the

unknown information MS_{key_j}. On the other hand, the $\mathcal{A}$ cannot compute R_l using L_p due to the unknown information PS_{key_k}. In the same way, the $\mathcal{A}$ cannot find the information ID_p. Therefore, the $\mathcal{A}$ could not get any confidential information even if he/she has the message $\langle L_p, N_p, O_p, ID_j, T_s, TID_p \rangle$.

- **Case 4.** Similar to Case 3, the $\mathcal{A}$ attempts to find private information used in the protocol using the public messages $\langle P_p, Q_p, S_p, T_k \rangle$ and $\langle M_2, M_3 \rangle$, where $P_p = h(ID_p \parallel SK \parallel ID_{pk} \parallel T_k \parallel h(ID_p \parallel PS_{key_k}))$, $Q_p = R_l'' \oplus h(ID_p \parallel PS_{key_k})$, $S_p = SK \oplus h(ID_p \parallel R_l'' \parallel T_k)$ and $M_2 = TID_p' \oplus R_l$, $M_3 = h(ID_p \parallel TID_p' \parallel r_l)$. The objective of the $\mathcal{A}$ is to find the private information $\langle ID_p, SK, PS_{key_k}, R_l \rangle$ from $\langle P_p, Q_p, S_p, T_k \rangle$ and $\langle M_2, M_3 \rangle$. Since all the public information are protected by the hash function $h(\cdot)$, it is extremely difficult to extract the confidential information $\langle ID_p, SK, PS_{key_k}, R_l \rangle$. Therefore, our protocol is secure even if the messages $\langle P_p, Q_p, S_p, T_k \rangle$ and $\langle M_2, M_3 \rangle$ are known to the $\mathcal{A}$
- **Case 5.** We suppose that the $\mathcal{A}$ captures the messages $\langle Y_1, Y_2, T_1, ID_{mj} \rangle$ and $\langle Y_3, Y_4, T_2 \rangle$, where $Y_1 = ID_p \oplus h(ID'_{mj} \parallel T_1 \parallel MK_{shrd})$ and $Y_2 = h(ID_{mj} \parallel ID'_{mj} \parallel MK_{shrd} \parallel T_1)$ and $Y_3 = h(ID_{pk} \parallel ID_p \parallel T_1 \parallel PS_{key_j})$, $Y_4 = ID_p \oplus h(ID_{pk} \parallel PS_{key_j} \parallel T_1)$. All the private information ID_p, MK_{shrd}, and PS_{key_j} are protected by the $h(\cdot)$. Therefore, our protocol is secured even if the messages $\langle Y_1, Y_2, T_1, ID_{mj} \rangle$ and $\langle Y_3, Y_4, T_2 \rangle$ are known to the $\mathcal{A}$.
- **Case 6.** We suppose that the $\mathcal{A}$ captures the message $\langle Y_5, Y_6, T_3 \rangle$, where $ID'_p = Y_4 \oplus h(ID_{pk} \parallel PS_{key_j} \parallel T_1)$, $Y'_3 = h(ID_{pk} \parallel ID'_p \parallel T_1 \parallel PS_{key_j})$. All the parameters of $\langle Y_5$, Y_6, $T_3 \rangle$ are protected due to the hardness of $h(\cdot)$, so it is infeasible by the $\mathcal{A}$ to compute all the private information $\langle SK, PS_{key_j}, ID_p \rangle$.

B. Discussion on Security Attacks of Our Protocol

In this section, we show that our proposed protocol is secure against all the known security threats. For the Case 1, we have demonstrated that the adversary could not get any advantage even if obtain all the information stored in the smartcard. Therefore, our protocol is secure against smartcard stolen attack. Moreover, the adversary cannot guess patient's secret information (low-entropy)such as identity, password, and biometric from the smartcard information. Therefore, the protocol is protected against the off-line password guessing attack.

We have also noticed in all the situations described in Case 1 to Case 6 that the adversary is unable to retrieve patient's identity ID_p. Hence, the anonymity of the patient is strongly achieved in our protocol.

In the patient registration phase, the patient only sends his/her ID_p to the medical-registration server MS_j. No information related to the password PW_p is sent to the MS_j, and further, the server cannot obtain PW_p from the known information. Hence, our protocol is free from insider attack.

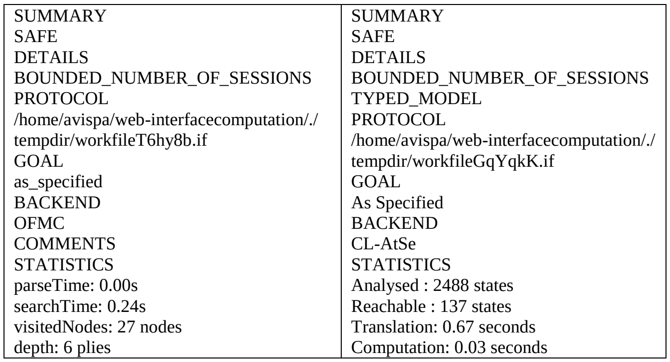
According to [17], the protocol may suffer from impersonation attack provided the adversary can extract some useful private information from the transmitted public messages. We have described Case 1 to Case 6 that the adversary cannot find the private information from the public messages. Therefore, our protocol is robust against patient impersonation attack, medical-server impersonation attack and physician-server impersonation attack.

In our protocol, we have used a random number SK as a session key, which is negotiated between the entities involved in the protocol. According to our analysis (see the Section V-A), the parameter SK is strongly protected. Hence, our protocol is free from session key security attack. Since we have utilized the concept of the timestamp in our protocol and each entity check the timestamp, so the adversary cannot launch the replay attack.

In the login phase of our protocol, the smartcard of the patient U_p first verifies the patient's biometric template and further verifies the identity and password. Afterwards, the smartcard sends the login message to the MS_j. Therefore, the unauthorized patient cannot generate a valid login message without. Moreover, an authorized patient cannot also login to the medical-server using wrong login credentials. Therefore, if the patient inputs wrong login credentials, the smartcard will detect and notify the patient. This kind of verification reduces communication and computation cost. Hence, our protocol is efficient. On the other hand, each entity performs some operations after verifying the received message from other entity. Therefore, the mutual authentication between all the entities is also satisfied in our protocol.

C. Protocol Simulation using AVISPA Software

In this section, we will present the protocol simulation protocol using AVISPA (see [22] for more information about the tool). We omit the inclusion of the HLPSL code for AVISPA, and only present the simulation findings for both OFMC and Cl-AtSe models (see Fig. 8). The findings show that our protocol is SAFE, which implies that the protocol is secure against active and passive attacks, replay attacks and man-in-the-middle attacks.



```
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/avispa/web-interfacecomputation/./
tempdir/workfileT6hy8b.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.24s
visitedNodes: 27 nodes
depth: 6 plies
```

(a) OFMC Results

```
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
TYPED_MODEL
PROTOCOL
/home/avispa/web-interfacecomputation/./
tempdir/workfileGqYqkK.if
GOAL
As Specified
BACKEND
CL-AtSe
STATISTICS
Analysed : 2488 states
Reachable : 137 states
Translation: 0.67 seconds
Computation: 0.03 seconds
```

(b) CL-AtSe Results

Fig. 8. Simulation results in OFMC and CL-AtSe backends

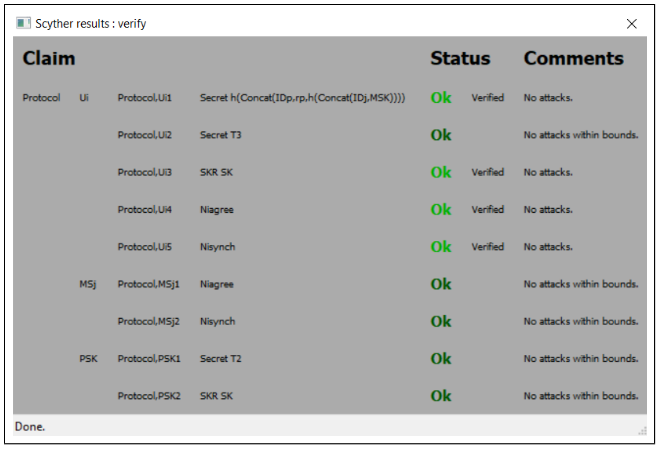
D. Protocol Simulation using Scyther Software

There are a number of protocol verification and simulation tools, such as ProVerif [10] used to ensure protocol correctness, AVISPA [11] to find potential protocol vulnerabilities, and Scyther [12], [13]. The latter is an automated tool, which uses pattern refinement to identify instances of falsification. Some key features of Scyther include:

- support for multi-protocol analysis;
- possibility of unbounded verification with guaranteed termination; and
- analysis of infinite sets of traces in terms of patterns.

Scyther takes as input a protocol description that includes a specification of the intended security properties (also known as security claims) for evaluation. The result window shows a summary of the claims in the protocol and the verification results. Here, one can find whether the protocol is correct or false. The tool has been successfully applied in studies such as those of [15], [16].

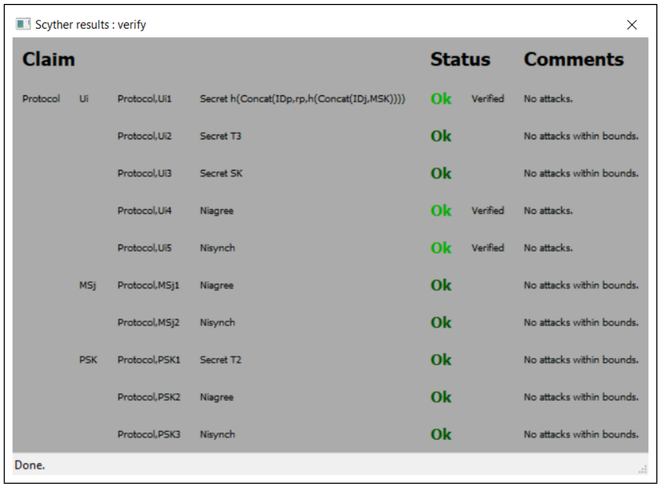
Figures 9 and 10 present the outputs from the security analysis of the proposed protocol using Scyther.



```
Scyther results : verify

Claim                                                                        Status   Comments
Protocol  Ui   Protocol,Ui1   Secret h(Concat(IDp,rp,h(Concat(IDj,MSK))))   Ok   Verified   No attacks.
               Protocol,Ui2   Secret T3                                      Ok              No attacks within bounds.
               Protocol,Ui3   SKR SK                                         Ok   Verified   No attacks.
               Protocol,Ui4   Niagree                                        Ok   Verified   No attacks.
               Protocol,Ui5   Nisynch                                        Ok   Verified   No attacks.
          MSj  Protocol,MSj1  Niagree                                        Ok              No attacks within bounds.
               Protocol,MSj2  Nisynch                                        Ok              No attacks within bounds.
          PSK  Protocol,PSK1  Secret T2                                      Ok              No attacks within bounds.
               Protocol,PSK2  SKR SK                                         Ok              No attacks within bounds.
Done.
```

Fig. 9. Security analysis using Scyther for Case 1



```
Scyther results : verify

Claim                                                                        Status   Comments
Protocol  Ui   Protocol,Ui1   Secret h(Concat(IDp,rp,h(Concat(IDj,MSK))))   Ok   Verified   No attacks.
               Protocol,Ui2   Secret T3                                      Ok              No attacks within bounds.
               Protocol,Ui3   Secret SK                                      Ok              No attacks within bounds.
               Protocol,Ui4   Niagree                                        Ok   Verified   No attacks.
               Protocol,Ui5   Nisynch                                        Ok   Verified   No attacks.
          MSj  Protocol,MSj1  Niagree                                        Ok              No attacks within bounds.
               Protocol,MSj2  Nisynch                                        Ok              No attacks within bounds.
          PSK  Protocol,PSK1  Secret T2                                      Ok              No attacks within bounds.
               Protocol,PSK2  Niagree                                        Ok              No attacks within bounds.
               Protocol,PSK3  Nisynch                                        Ok              No attacks within bounds.
Done.
```

Fig. 10. Security analysis using Scyther for Case 2

VI. Performance Study

This section discusses the performance of the proposed protocol with related protocols in terms of different security attacks, computation cost, communication cost, and smartcard storage cost. We compare our protocol with Amin and Biswas [17], Das et al. [18], Maitra et al. [24], Li et al. [25], and Wang and Ma [26] as these protocols use the similar architecture and execute similar operations operations. Several phases are involved in these protocol, but we did not consider the patient registration phase and medical-server registration phase because these phases are executed only one time. In addition to that, the use of password change phase depends on user choice and thus it is also excluded from the comparison.

Note that user anonymity is one of the the most important security attributes, especially to design a secure and robust TMIS system. Besides, protecting user traceability is also equally important. In Das et al.'s protocol [18], they describe that Amin and Biswas [17] protocol is not secure for the several reasons such as insider attack, replay and man-in-the-middle attacks, and design flaws in the registration, login and authentication phases. In this paper, we demonstrate that Das et al.'s protocol does not preserve user anonymity and traceability property. In addition, the same protocol is also vulnerable to smartcard launch attack. Therefore, Das et al.'s protocol is not completely secure, it is not applicable for real-life applications. We then design an authentication protocol and its security analysis claim that the protocol is completely secure including anonymity and traceability under the hardness assumption of the hash function.

TABLE II
Computation cost comparison

Protocol	CL	CA	CP	TC
Amin and Biswas [17]	$5T_h$	$14T_h$	$6T_h$	0.0100
Das et al. [18]	$5T_h$	$15T_h$	$8T_h$	0.0110
Maitra et al. [24]	$4T_h + T_e + T_s$	$6T_h + T_s$	$6T_h$	2.0900
Li et al. [25]	$2T_h$	$25T_h$	$7T_h$	0.0130
Wang and Ma [26]	$4T_h + 2T_s$	$7T_h + 4T_s$	$4T_h + 2T_s$	1.0500
Proposed	$5T_h$	$(16T_h \mid 11T_h)$	$5T_h$	0.0084

CL: Computation cost for login phase, **CA**: Computation cost for authentication phase, **CP**: Computation cost for password change phase, **TC**: Total execution cost (ms)

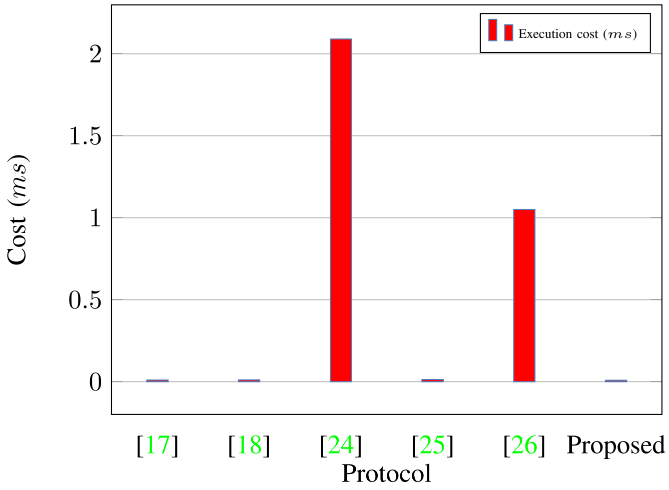


Fig. 11. Total execution cost of different protocols

Now, we evaluate the computation cost of our proposed protocol and existing protocols [17], [18], [24], [25], [26]. Table II and presents the login, authentication, and password change phases. The earlier authentication protocols are vulnerable to different attacks. We found from Table II that our protocol takes almost same computation cost in comparison with earlier protocols. Note that our protocol protects all possible security threats by executing the almost same number of hash operation (T_h). It is also noted that our protocol additionally achieves user traceability and anonymity.

We use the experimental results of [23], where the SHA-1 hash function has been executed using MIRACL C/C++ Library. The system uses 32-bit Windows 7 OS, Visual C++ 2008 S/W. According to [23], the SHA-1 hash function takes $T_h \approx 0.0004$ milliseconds, AES symmetric encryption takes $T_s \approx 0.1303$ milliseconds and the exponentiation operation takes $T_e \approx 1.8269$ milliseconds. We have also used others Ref. [32], [33], [34] for runtime verification. From Fig. 12 and Table II, it is clear that the protocols in [24], [26] need huge time whereas ours and [18], [25] require very less time.

TABLE III
SMARTCARD STORAGE COST AND COMMUNICATION COST

Protocol	**SC** (bit)	**LC** (bit)	**AC** (bit)
Amin and Biswas [17]	$384 + (3 \times 128 \times n)$	768	1152
Das et al. [18]	$768 + (3 \times 128 \times n)$	480	1098
Maitra et al. [24]	$512 + (2 \times 128 \times n)$	512	384
Li et al. [25]	768	512	1664
Wang and Ma [26]	384	320	256
Proposed	896	640	(1536 | 1280)

SC: Smartcard storage cost, **LC**: Communication cost for login phase, **AC**: Communication cost for authentication phase, **CL**: Computation cost for login phase, **CA**: Computation cost for authentication phase, **CP**: Computation cost for password change phase, n: Total number of medical-servers

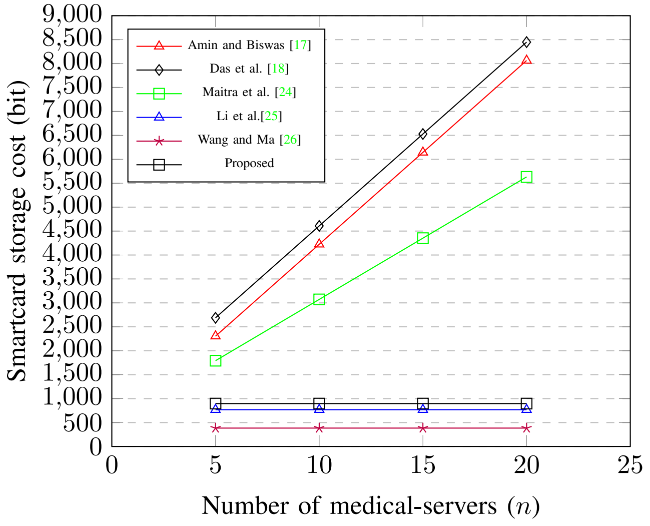


Fig. 12. Smartcard storage cost of different protocols

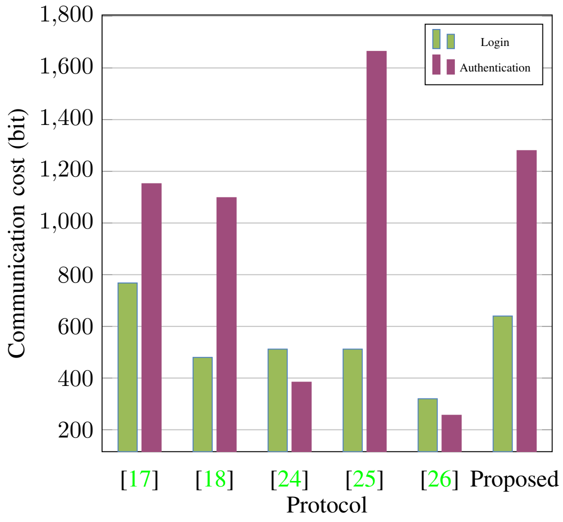


Fig. 13. Communication cost of login and authentication phases of different protocols

We evaluate the smartcard storage cost and communication cost of our protocol including previous works in Table III. The storage area in a smartcard is very limited. We found in Table III that the smartcard storage cost in the protocols [17], [18], [24] varies linearly with the number of medical-servers (n) whereas our protocol, Li et al.'s protocol [25] and, Wang and Ma [26] are independent of n. Because, the architecture of their protocol is different. They have only considered multiple number of servers, whereas the protocols in [17], [18], [24] and ours considered multi-medical systems with multiple number of medical-servers. The major contribution of our protocol is that it is independent of n though it is based on the multiple medical-servers. The pictorial representation of the smartcard storage cost of different protocols including ours are presented in Fig. 12, which shows that our protocol is very efficient compared to the protocols in [17], [18], [24], [25]. In Fig. 13, we also present the communication cost (bit) for login and authentication phases of our protocol and the protocols in [17], [18], [24], [25], [26]. Our protocol is very efficient in terms of communication cost compared to the protocols in [17], [18], [24], [26]. In order to resist the user traceability attack, our protocol incurs additional one communication round. Hence, our protocol takes slightly more communication cost than the existing protocols.

VII. CONCLUSION

Designing secure cryptographic protocols remains challenging, as demonstrated in this paper (i.e. revealing insecurities in Das et al.'s protocol). More importantly, we proposed an efficient and reliable user authentication protocol for the TMIS environment. Findings from the security and performance evaluations revealed that our protocol is secure against common security threats and provides both user anonymity and user untraceability, as well as being efficient in comparison to related protocols.

Future research includes collaborating with a real-world healthcare organization to implement a prototype of the proposed protocol within a division or department. This will allow us to evaluate its security and performance in a real-world setting, and revise its design if necessary.

REFERENCES

[1] https://innovatemedtec.com/digital-health/ehealth, last accessed 26 March 2018

[2] Meng, W., Choo, K.-K. R., Furnell, S., Vasilakos, A. V., & Probst, C. W. (2018). “Towards Bayesian-based Trust Management for Insider Attacks in Healthcare Software-Defined Networks”, IEEE Transactions on Network and Service Management, in press; DOI: 10.1109/TNSM.2018.2815280.

[3] Chuang, M. C., & Chen, M. C. (2014). “An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics”, Expert Systems with Applications, 41(4), 1411-1418.

[4] Li, Xiong, Jianwei Niu, Md Zakirul Alam Bhuiyan, Fan Wu, Marimuthu Karuppiah, and Saru Kumari. ”A Robust ECC based Provable Secure Authentication Protocol with Privacy Protection for Industrial Internet of Things.” IEEE Transactions on Industrial Informatics (2017).

[5] Shen, Jian, Ziyuan Gui, Sai Ji, Jun Shen, Haowen Tan, and Yi Tang. ”Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks.” Journal of Network and Computer Applications (2018).

[6] Rahman, Farzana, Md Zakirul Alam Bhuiyan, and Sheikh Iqbal Ahamed. ”A privacy preserving framework for RFID based healthcare systems.” Future Generation Computer Systems 72 (2017): 339-352.

[7] He, D., Kumar, N., Chen, J., Lee, C. C., Chilamkurti, N., & Yeo, S. S. (2015). “Robust anonymous authentication protocol for healthcare applications using wireless medical sensor networks”, Multimedia Systems, 21(1), 49-60. Chicago

[8] Mishra, D., Das, A. K., & Mukhopadhyay, S. (2014). “A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. Expert Systems with Applications”, 41(18), 8129-8143.

[9] Gope, P., & Hwang, T. (2016). “A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks”, IEEE Transactions on Industrial Electronics, 63(11), 7124-7132.

[10] Blanchet, Bruno. “An Efficient Cryptographic Protocol Verifier Based on Prolog Rules”, csfw. Vol. 1. 2001.

[11] Armando, Alessandro, et al. “The AVISPA tool for the automated validation of internet security protocols and applications”, International conference on computer aided verification. Springer, Berlin, Heidelberg, 2005.

[12] Cremers, Cas JF. “The scyther tool: Verification, falsification, and analysis of security protocols”, CAV. Vol. 8. 2008.

[13] Cremers, Cas JF. “Scyther: Semantics and verification of security protocols. Eindhoven”, Netherlands: Eindhoven University of Technology, 2006.

[14] Gope, Prosanta, and Ruhul Amin. ”A novel reference security model with the situation based access policy for accessing ephr data.” Journal of medical systems 40, no. 11 (2016): 242.

[15] Ali, Mazhar, Saif Malik, and Samee Khan. “DaSCE: Data security for cloud environment with semi-trusted third party”, IEEE Transactions on Cloud Computing (2015).

[16] Cao, Jin, Maode Ma, and Hui Li. “G2RHA: Group-to-Route Handover Authentication Scheme for Mobile Relays in LTE-A High-Speed Rail Networks”, IEEE Transactions on Vehicular Technology 66.11 (2017): 9689-9701.

[17] Amin, R., & Biswas, G. P. (2015). “A novel user authentication and key agreement protocol for accessing multi-medical server usable in tmis”, Journal of medical systems, 39(3), 33.

[18] Das, A. K., Odelu, V., & Goswami, A. (2015). “A secure and robust user authenticated key agreement scheme for hierarchical multi-medical server environment in TMIS”, Journal of medical systems, 39(9), 92.

[19] Lin, Hao, Fengtong Wen, and Chunxia Du. ”An Improved User Authentication and Key Agreement Scheme for Multi-medical Server Usable in TMIS.” DEStech Transactions on Engineering and Technology Research iect (2016).

[20] Liu, X., Li, Y., Qu, J. and Ding, Y., 2017. A Lightweight Pseudonym Authentication and Key Agreement Protocol for Multi-medical Server Architecture in TMIS. KSII Transactions on Internet & Information Systems, 11(2).

[21] Amin, R., & Biswas, G. P. (2016). “A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks”, Ad Hoc Networks, 36, 58-80.

[22] Armando, A. et al., “The AVISPA tool for the automated validation of internet security protocols and applications”, In: Proceedings of the 17th International conference on computer aided verification, LNCS, vol. 3576, pp. 281285, 2005.

[23] Xu, L., Wu, F., ”Cryptanalysis and Improvement of a User Authentication Scheme Preserving Uniqueness and Anonymity for Connected Health Care”, Journal of Medical Systems, 39: 10, 2015. DOI 10.1007/s10916-014-0179-x

[24] Maitra, T., and Giri, D., “An efficient biometric and passwordbased remote user authentication using smart card for telecare medical information systems in multi-server environment”. J. Med. Syst. 38(12):142, 2014. doi:10.1007/s10916-014-0142-x.

[25] X.Li, Y.P.Xiong, J.Ma, W.D.Wang, “An efficient and security dynamic identity based authentication protocol for multi-server architecture using smartcards”, Journal of Network and Computer Applications 35(2) (2012)763-769.

[26] B.Wang, and M. Ma, “A Smart Card Based Efficient and Secured Multi-Server Authentication Scheme”, Wireless Personal Communications, vol. 68, no. 2, pp. 361-378, 2013.

[27] S.H. Islam, A.K. Das, M.K. Khan. “An efficient biometric-based password authentication scheme for client-server environment using ECC and fuzzy extractor”, International Journal of Ad Hoc and Ubiquitous Computing, 2015. DOI: 10.1504/IJAHUC.2015.10001794

[28] S.H. Islam, A.K. Das, M.K. Khan. “Design of a provably secure identity-based digital multi-signature scheme using biometrics and fuzzy extractor”, Security and Communication Networks, 9(16), 3229-3238, 2016.

[29] X. Li, J-W. Niu, J. Ma, W-D. Wang, C-L. Liu. Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. Journal of Network and Computer Applications, 34(1):73-79, 2011.

[30] P. Kocher, J. Jaffe, and B. Jun, Differential power analysis, Proceedings of the Advances in Cryptology(CRYPTO’99), LNCS, vol. 1666, pp. 388-397, 1999.

[31] T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smartcard security under the threat of power analysis attacks. IEEE Transactions on Computers, 51(5):541-552, 2002.

[32] Jiang, Yu, Houbing Song, Rui Wang, Ming Gu, Jiaguang Sun, and Lui Sha. ”Data-centered runtime verification of wireless medical cyber-physical system.” IEEE transactions on industrial informatics 13, no. 4 (2017): 1900-1909.

[33] Ou, Andrew Y-Z., Maryam Rahmaniheris, Yu Jiang, Po-Liang Wu, and Lui Sha. ”Toward safe interoperations in network connected medical cyber-physical systems using open-loop safe protocols.” In Proceedings of the 36th International Conference on Computer-Aided Design, pp. 957-963. IEEE Press, 2017.

[34] Jiang, Yu, Mingzhe Wang, Han Liu, Mohammad Hosseini, and Jiaguang Sun. ”Dependable integrated clinical system architecture with runtime verification.” In Computer-Aided Design (ICCAD), 2017 IEEE/ACM International Conference on, pp. 951-956. IEEE, 2017.

[35] Chunyong Yin, Jinwen Xi, Ruxia Sun, Jin Wang. ”Location Privacy Protection Based on Differential Privacy Strategy for Big Data in Industrial Internet of Things .” IEEE Transactions on Industrial Informatics, 14(8), pp. 3628-3636, 2017.

[36] Ping Guo, Jin Wang, Sai Ji, Xue Hua Geng, Neal N. Xiong. ”A Lightweight Encryption Scheme Combined with Trust Management for Privacy-Preserving in Body Sensor Networks”. Journal of Medical Systems, 39(12), pp.190-198, 2015.

[37] Jian Shen, Haowen Tan, Sangman Moh, Ilyong Chung, Jin Wang. ”An Efficient RFID Authentication Protocol Providing Strong Privacy and Security”, Journal of Internet Technology, 17(3), 2016

Ruhul Amin received Ph.D. in Computer Science and Engineering from the Indian Institute of Technology [IIT(ISM)] Dhanbad, Jharkhand, India, in 2017. He also received B.Tech. and M.Tech. both in Computer Science and Engineering from Maulana Abul Kalam Azad University of Technology, West Bengal, India in 2009 and 2013, respectively. Presently, he is working as an Assistant Professor in the Department of Computer Science and Engineering, Dr. Shyama Prasad Mukherjee International Institute of Information Technology, Naya Raipur, India. His research interest includes authentication protocol and security in WSNs.

SK Hafizul Islam (SM'18) received the M.Sc. degree in Applied Mathematics from Vidyasagar University, Midnapore, India, in 2006, and the M.Tech. degree in Computer Application and the Ph.D. degree in Computer Science and Engineering in 2009 and 2013, respectively, from the Indian Institute of Technology [IIT (ISM)] Dhanbad, Jharkhand, India, under the INSPIRE Fellowship Ph.D. Program (funded by Department of Science and Technology, Government of India). He is currently an Assistant Professor with the Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani (IIIT Kalyani), West Bengal, India. Before joining the IIIT Kalyani, he was an Assistant Professor with the Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani (BITS Pilani), Rajastahn, India. He has more than five years of teaching and eight years of research experiences. He has authored or co-authored seventy five research papers in journals and conference proceedings of international reputes. His research interests include cryptography, information security, WSNs, IoT, and cloud computing. Dr. Islam is an Associate Editor for Wiley's ***International Journal of Communication Systems*** and ***Security and Privacy***. He was a reviewer in many reputed international journals and conferences. He was the recipient of the University Gold Medal, the S.D. Singha Memorial Endowment Gold Medal, and the Sabitri Parya Memorial Endowment Gold Medal from Vidyasagar University, in 2006. He was also the recipient of the University Gold Medal from IIT(ISM) Dhanbad in 2009 and the OPERA award from BITS Pilani in 2015. He is a member of IEEE and ACM.

Prosanta Gope received the M.Tech. degree in Computer Science and Engineering from the National Institute of Technology (NIT), Durgapur, India, in 2009, and the Ph.D. degree in Computer Science and Information Engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 2015. He is currently working as a Lecturer in the Department of Computer Science (Cyber Security) at the University of Hull. Prior to this, Dr. Gope was working as a Research Fellow in the Department of Computer Science at National University of Singapore (NUS). His research interests include lightweight authentication, authenticated encryption, access control system, security in mobile communication and cloud computing and hardware security of the IoT devices. He has authored over 50 peer-reviewed articles in several reputable international journals and conferences, and has four filed patents. He received the Distinguished Ph.D. Scholar Award in 2014 from the National Cheng Kung University, Tainan, Taiwan. He is currently an editor of the *Mobile Information Systems journal*.

Kim-Kwang Raymond Choo received the Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio, and an adjunct associate professorship at the University of South Australia. He serves on the editorial board of Computers & Electrical Engineering, Cluster Computing, Digital Investigation, IEEE Access, IEEE Cloud Computing, IEEE Communications Magazine, Future Generation Computer Systems, Journal of Network and Computer Applications, PLoS ONE, Soft Computing, etc. He also serves as the Special Issue Guest Editor of ACM Transactions on Embedded Computing Systems (2017), ACM Transactions on Internet Technology (2016), Computers and Electrical Engineering (2017), Digital Investigation (2016), Future Generation Computer Systems (2016,2018), IEEE Cloud Computing (2015), IEEE Network (2016), IEEE Transactions on Cloud Computing (2017), IEEE Transactions on Dependable and Secure Computing (2017), Journal of Computer and System Sciences (2017), Multimedia Tools and Applications (2017), Personal and Ubiquitous Computing (2017), Pervasive and Mobile Computing (2016), Wireless Personal Communications (2017), etc. In 2016, he was named the Cybersecurity Educator of the Year APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn), and in 2015 he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen-Nuremberg. He is the recipient of ESORICS 2015 Best Paper Award, 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, Fulbright Scholarship in 2009, 2008 Australia Day Achievement Medallion, and British Computer Society's Wilkes Award in 2008. He is also a Fellow of the Australian Computer Society, and a Senior Member of IEEE.

Nachiket Tapas received the M.Tech. degree in computer science and engineering from the National Institute of Technology Goa, India, in 2016. He is currently persuing Ph.D. from the Department of Engineering, University of Messina, Italy 98158. Prior to this, Mr. Tapas served more than four year as a Senior Software Engineer at Infosys Technologies Limited, Bangalore. His research interests include Internet of Things, Data Mining, and security in mobile communication and cloud computing.