IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 23, 2021, accepted May 7, 2021, date of publication May 11, 2021, date of current version May 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3079312

On the Security of a Secure and Lightweight Authentication Scheme for Next Generation IoT Infrastructure

ASHOK KUMAR DAS[1], (Senior Member, IEEE), BASUDEB BERA[1], MOHAMMAD WAZID[2], (Senior Member, IEEE), SAJJAD SHAUKAT JAMAL[3], AND YOUNGHO PARK[4], (Member, IEEE)

[1]Center for Security, Theory, and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India
[2]Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248002, India
[3]Department of Mathematics, King Khalid University, Abha 61413, Saudi Arabia
[4]School of Electronics Engineering, Kyungpook National University, Daegu 41566, South Korea

Corresponding authors: Ashok Kumar Das (iitkgp.akdas@gmail.com) and Youngho Park (parkyh@knu.ac.kr)

This work was supported in part by the Deanship of Scientific Research at King Khalid University through research groups program under Grant R.G.P. 2/48/42, and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1I1A3058605.

ABSTRACT In recent years, the Internet of things (IoT) has become an encouraging communication paradigm that has numerous applications including smart city, smart home and intelligent transportation system. The information sensed by several IoT smart devices can be security stored at the (cloud) servers. An external user, being a client, can access the services from a server for the sensing information, provided that a mutual authentication happens among them. Using the established session key among the user and the server, encrypted information with the help of session key can be delivered to the user by the server securely. Recently, Rana *et al.* proposed a smart-card based remote user authentication scheme using user password. In this comment paper, we carefully analyzed the scheme of Rana *et al.* and tracked down that their scheme is insecure against serious attacks, including stolen smart card attack, privileged-insider attack, user impersonation attack, password change attack and Ephemeral Secret Leakage (ESL) attack. Furthermore, their scheme does not preserve untraceability feature. To remedy these security pitfalls, we also provide some remedies that can help in building more secure and effective user authentication scheme to apply in securing next generation IoT infrastructure.

INDEX TERMS Internet of Things (IoT), cryptanalysis, authentication, key agreement, security.

I. INTRODUCTION

In recent years, the Internet of things (IoT) has become an encouraging communication paradigm. IoT contains various types of devices, like sensors, microcontrollers, and transceivers that can be applied for an effective system. If we make comparison of the IoT services offered under the 5G (5th generation mobile network) deployment, 6G (6th generation mobile network) IoT has the capability to offer high-density heterogeneous types of smart devices which are involved for high capacity, more robust system architecture support and smart algorithms using the Artificial Intelligence (AI) [1]. Due to huge deployment of IoT smart devices, while the Big data analytics become more essential, at the same time maintaining the security among the IoT devices and the deployed gateway nodes is also becoming challenging task. Access control and authentication are two important security services to secure different networking environments [2]–[11].

In a smart card based remote user authentication, an authorized registered user and a remote server need to authenticate each other in order to make secure communication. After mutual authentication, both the communicating parties establish a session key which can be further used to secure communication among them for accessing the services from a remote server by a legal user. Starting from the seminal work designed by Lamport [12] in 1981, several remote user authentication mechanisms have been proposed in the

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

literature [13]–[19]. However, the major of these schemes are inefficient for practical implementations or they are vulnerable to various potential attacks, such as privileged-insider attack, stolen smart card attack, replay and man-in-the-middle attacks, impersonation attacks, and so on. Later, in order to strengthen the security of a smart-card based remote user authentication, user biometric plays an important role in designing biometric-based authentication schemes [20], [21].

In 2016, Kaul and Awasthi designed a smart-card based remote user authentication scheme [22] in which a user being a client can authenticate with a remote server with the help of the credentials stored in his/her smart card. However, recently, in 2021, Rana *et al.* [23] reviewed the scheme of Kaul and Awasthi, and pointed out the vulnerability to user impersonation attack in Kaul and Awasthi's scheme. In order to remedy such security weakness, they suggested an improved solution and claimed that their scheme is successfully defended the security problem found in Kaul and Awasthi's scheme. In this work, we carefully analyze the scheme of Rana *et al.* and show that their design led to reveal not only user impersonation attack, but also other attacks that are mentioned in Section I-A.

A. RESEARCH CONTRIBUTIONS

The following are the primary contributions:

- We define a threat model which provides various capabilities of a passive or an active adversary.
- We then critically analyze a recently proposed Rana *et al.*'s scheme [23] and show that this scheme is unfortunately designed with several serious security weaknesses. In particular, we show that their scheme cannot resist stolen smart card attack, privileged-insider attack, user impersonation attack, password change attack and "Ephemeral Secret Leakage (ESL)" attack. Moreover, we show that their scheme fails to provide untraceability feature, which is a very important feature in a user authentication protocol.
- Next, we suggest some remedies that can be applied to overcome the security pitfalls found in Rana *et al.*'s scheme.

B. PAPER OUTLINE

The sketching of this paper is organized as follows. In the next section, an attack model has been discussed. In Section III, we review a recently proposed Rana *et al.*'s scheme [23], and then provide its detailed cryptanalysis in Section IV. Some remedies are discussed in Section V to overcome the security pitfalls and design flaws found in Rana *et al.*'s scheme. The paper is then wound up in Section VI.

II. ATTACK MODEL

In the considered attack model, we consider the following capabilities of an adversary:

- We contemplate the widely-recognized "Dolev and Yao threat model (also, known as DY model)" [24]. This model permits two communicating participants to communicate over an insecure (public) channel. Thus, an adversary $\mathcal{AE}$ has full control of the communication channel, where it can not only eavesdrop(read) the messages, but also can modify, erase or insert fake messages contents, during the communication. In addition, the end-point entities (such as users) are not trusted in common.
- We contemplate another *de factor* adversary model, known as the "Canetti and Krawczyk adversary model (known as CK-adversary model)" [25]. A CK-adversary $\mathcal{AE}$ retains the same capability of an adversary under the DY model. In addition, $\mathcal{AE}$ can compromise the session states and private keys through the session-hijacking attacks.
- Using the revolutionary power analysis attacks [26], an adversary $\mathcal{AE}$ can obtain all the sensitive credentials stored in a lost stolen) smart card of a valid registered user in the network. The extracted credentials can be further used to launch other attacks, like privileged-insider, user impersonation, password change and "Ephemeral Secret Leakage (ESL)" attacks.

III. REVIEW OF RANA *et al.*'s SCHEME

In this section, we review the recently proposed Rana *et al.*'s scheme [23] in order to show its various security pitfalls in Section V. To discuss the Rana *et al.*'s scheme, a list of notations and their significance is provided in Table 1.

TABLE 1. Symbols used in the paper.

Symbol	Its significance
Usr_i	i-th user
ID_{Usr_i}, Pwd_{Usr_i}	Identity and password of Usr_i, respectively
m_{Usr_i}	Random secret of Usr_i
RPW_{Usr_i}	Pseudo-password of Usr_i
SC_{Usr_i}	Smart card of Usr_i
$\oplus, \|\|$	Bitwise XOR and string concatenation operations, respectively
S	Server
TS_{Usr_i}, TS_S	Timestamps generated by Usr_i and S, respectively
δTS	Maximum allowable transmission delay
K_S	Secret key of S
$Enc_K(\cdot)/Dec_K(\cdot)$	Symmetric encryption/decryption using key K
$CHash(\cdot)$	Collision-resistant one-way cryptographic hash function
$\rightarrow$	A public (insecure) channel
$\Rightarrow$	A protected (secure) channel
SK	Session key between Usr_i and S
$\mathcal{AE}$	Passive/active adversary

A. REGISTRATION PHASE

In order to register a user Usr_i to the remote server S, the following steps need to be executed via secure channel. Note that the registration process is one-time process and it can be also done in offline (secure) mode.

- Step Reg_1: The user Usr_i has the freedom of selecting his/her own identity and password. Let Usr_i pick

User (Usr_i)	Server (S)
Input ID_{Usr_i} and Pwd_{Usr_i} Generate random secret m_{Usr_i} Compute $RPW_{Usr_i} = CHash(m_{Usr_i} \|\| Pwd_{Usr_i})$, $RReq = \{ID_{Usr_i}, RPW_{Usr_i}\}$ $\Longrightarrow$	
	Pick random secret y_{Usr_i}, own secrets a and b Calculate $\overline{DID_{Usr_i}} = Enc_{K_S}[ID_{Usr_i} \|\| y_{Usr_i}]$, $\alpha_{Usr_i} = CHash((ID_{Usr_i} \oplus a) \|\| b)$, $\beta_{Usr_i} = \alpha_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus RPW_{Usr_i})$, $\gamma_{Usr_i} = y_{Usr_i} \oplus CHash(\alpha_{Usr_i} \oplus RPW_{Usr_i})$, $\psi_{Usr_i} = CHash(ID_{Usr_i} \|\| RPW_{Usr_i} \|\| y_{Usr_i} \|\| \alpha_{Usr_i})$ Insert $\{\beta_{Usr_i}, \gamma_{Usr_i}, \overline{DID_{Usr_i}}, \psi_{Usr_i}$, $CHash(\cdot)\}$ into a smart card SC_{Usr_i} $RRes = SC_{Usr_i}$ $\Longleftarrow$
Compute $\eta_{Usr_i} = m_{Usr_i} \oplus CHash(ID_{Usr_i} \|\| Pwd_{Usr_i})$ Now, $SC_{Usr_i} = \{\beta_{Usr_i}, \gamma_{Usr_i}$, $\overline{DID_{Usr_i}}, \psi_{Usr_i}, \eta_{Usr_i}, CHash(\cdot)\}$	

FIGURE 1. Summary of registration phase in Rana *et al.*'s scheme.

ID_{Usr_i} and Pwd_{Usr_i} as the identity and password, respectively. Next, Usr_i starts calculating pseudo-password as $RPW_{Usr_i} = CHash(m_{Usr_i} || Pwd_{Usr_i})$ after generating a random secret m_{Usr_i}. After that, Usr_i transmits the registration request $RReq = \{ID_{Usr_i}, RPW_{Usr_i}\}$ to the server S via a secure channel.

- Step Reg_2: After reception of $RReq$ from the user Usr_i, S picks another random secret y_{Usr_i} for Usr_i and its own two secrets a and b for computing the following components:

$$\begin{aligned}\overline{DID_{Usr_i}} &= Enc_{K_S}[ID_{Usr_i}||y_{Usr_i}],\\ \alpha_{Usr_i} &= CHash((ID_{Usr_i} \oplus a)||b),\\ \beta_{Usr_i} &= \alpha_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus RPW_{Usr_i}),\\ \gamma_{Usr_i} &= y_{Usr_i} \oplus CHash(\alpha_{Usr_i} \oplus RPW_{Usr_i}),\\ \psi_{Usr_i} &= CHash(ID_{Usr_i}||RPW_{Usr_i}||y_{Usr_i}\\ &\quad ||\alpha_{Usr_i}),\end{aligned}$$

  where K_S is the secret key of S which is used for symmetric encryption and decryption. S then inserts the information $\{\beta_{Usr_i}, \gamma_{Usr_i}, \overline{DID_{Usr_i}}, \psi_{Usr_i}, CHash(\cdot)\}$ into a smart card SC_{Usr_i} and sends the registration response $RRes = SC_{Usr_i}$ to Usr_i via secure channel.
- Step Reg_3: After receiving $RRes$, Usr_i calculates $\eta_{Usr_i} = m_{Usr_i} \oplus CHash(ID_{Usr_i} || Pwd_{Usr_i})$ and inserts it into SC_{Usr_i}.

At the end of this phase, $SC_{Usr_i} = \{\beta_{Usr_i}, \gamma_{Usr_i}, \overline{DID_{Usr_i}}, \psi_{Usr_i}, \eta_{Usr_i}, CHash(\cdot)\}$. This phase is also briefed in Figure 1.

B. LOGIN PHASE

Once a user Usr_i registers with the server S, he/she is ready to login in the system with the help of his/her own smart card $SC_{Usr_i} = \{\beta_{Usr_i}, \gamma_{Usr_i}, \overline{DID_{Usr_i}}, \psi_{Usr_i}, \eta_{Usr_i}, CHash(\cdot)\}$. The following steps are then essential:

https:orcid.org1∗

- Step Log_1: After inserting the smart card SC_{Usr_i}, the user Usr_i inputs his/her credentials, like the identity $ID^*_{Usr_i}$ and password $Pwd^*_{Usr_i}$. Then, SC_{Usr_i} calculates the following:

$$\begin{aligned} m_{Usr_i} &= \eta_{Usr_i} \oplus CHash(ID^*_{Usr_i}||Pwd^*_{Usr_i}),\\ RPW^*_{Usr_i} &= CHash(m_{Usr_i}||Pwd^*_{Usr_i}),\\ \alpha^*_{Usr_i} &= \beta_{Usr_i} \oplus CHash(ID^*_{Usr_i} \oplus RPW^*_{Usr_i}),\\ y^*_{Usr_i} &= \gamma_{Usr_i} \oplus CHash(\alpha^*_{Usr_i} \oplus RPW^*_{Usr_i}),\\ \psi^*_{Usr_i} &= CHash(ID^*_{Usr_i}||RPW^*_{Usr_i}||y^*_{Usr_i}\\ &\quad ||\alpha^*_{Usr_i}).\end{aligned}$$

  Next, SC_{Usr_i} checks the validity of $\psi^*_{Usr_i} = \psi_{Usr_i}$. If it holds, the login request of Usr_i is accepted by the server S. Otherwise, the phase is terminated here.
- Step Log_2: SC_{Usr_i} also calculates the following components in order to form an authentication request $AuthReq = \{\overline{DID_{Usr_i}}, \omega_{Usr_i}, \theta_{Usr_i}, TS_{Usr_i}\}$ by generating fresh timestamp TS_{Usr_i}:

$$\begin{aligned}\omega_{Usr_i} &= y^*_{Usr_i} \oplus CHash(ID^*_{Usr_i} \oplus \alpha^*_{Usr_i})\\ &\quad \oplus CHash(ID^*_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus TS_{Usr_i}),\\ \theta_{Usr_i} &= CHash(ID^*_{Usr_i}||\alpha^*_{Usr_i}||y^*_{Usr_i}||\\ &\quad (\alpha^*_{Usr_i} \oplus y^*_{Usr_i})||TS_{Usr_i}).\end{aligned}$$

User (Usr_i)/Smart Card (SC_{Usr_i})	Server (S)
Input identity $ID^*_{Usr_i}$ and password $Pwd^*_{Usr_i}$ Compute $m_{Usr_i} = \eta_{Usr_i} \oplus CHash(ID^*_{Usr_i} \Vert Pwd^*_{Usr_i})$, $RPW^*_{Usr_i} = CHash(m_{Usr_i} \Vert Pwd^*_{Usr_i})$, $\alpha^*_{Usr_i} = \beta_{Usr_i} \oplus CHash(ID^*_{Usr_i} \oplus RPW^*_{Usr_i})$, $y^*_{Usr_i} = \gamma_{Usr_i} \oplus CHash(\alpha^*_{Usr_i} \oplus RPW^*_{Usr_i})$, $\psi^*_{Usr_i} = CHash(ID^*_{Usr_i} \Vert RPW^*_{Usr_i} \Vert y^*_{Usr_i}$ $\Vert \alpha^*_{Usr_i})$ Check if $\psi^*_{Usr_i} = \psi_{Usr_i}$? If valid, generate fresh timestamp TS_{Usr_i} Compute $\omega_{Usr_i} = y^*_{Usr_i} \oplus CHash(ID^*_{Usr_i} \oplus \alpha^*_{Usr_i})$ $\oplus CHash(ID^*_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus TS_{Usr_i})$, $\theta_{Usr_i} = CHash(ID^*_{Usr_i} \Vert \alpha^*_{Usr_i} \Vert y^*_{Usr_i} \Vert$ $(\alpha^*_{Usr_i} \oplus y^*_{Usr_i}) \Vert TS_{Usr_i})$ $\xrightarrow{AuthReq = \{\overline{DID_{Usr_i}}, \omega_{Usr_i}, \theta_{Usr_i}, TS_{Usr_i}\}}$	
	Check if $(TS^*_{Usr_i} - TS_{Usr_i}) \leq \delta TS$? If so, extract $(ID_{Usr_i} \Vert y_{Usr_i}) = Dec_{K_s}[\overline{DID_{Usr_i}}]$ Calculate $\alpha^*_{Usr_i} = CHash((ID_{Usr_i} \oplus a) \Vert b)$, $y^*_{Usr_i} = \omega_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i})$ $\oplus CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus TS_{Usr_i})$, $\theta^*_{Usr_i} = CHash(ID_{Usr_i} \Vert \alpha^*_{Usr_i} \Vert y^*_{Usr_i} \Vert$ $(\alpha^*_{Usr_i} \oplus y^*_{Usr_i}) \Vert TS_{Usr_i})$ Check if $\theta^*_{Usr_i} = \theta_{Usr_i}$? If so, generate fresh timestamp TS_S Compute $\mu_{Usr_i} = CHash(ID_{Usr_i} \Vert y^*_{Usr_i} \Vert$ $(\alpha^*_{Usr_i} \oplus y^*_{Usr_i}) \Vert TS_S)$ $\xleftarrow{AuthRes = \{\mu_{Usr_i}, TS_S\}}$
Check validity of timestamp $(TS^*_S - TS_S) \leq \delta TS$ If valid, calculate $\mu^*_{Usr_i} = CHash(ID_{Usr_i} \Vert y^*_{Usr_i} \Vert$ $(\alpha^*_{Usr_i} \oplus y^*_{Usr_i}) \Vert TS_S)$ Verify if $\mu^*_{Usr_i} = \mu_{Usr_i}$? If so, compute session key shared with S as $SK = CHash(ID^*_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus y^*_{Usr_i}$ $\oplus TS_{Usr_i} \oplus TS_S)$	Compute session key shared with Usr_i as $SK = CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus y^*_{Usr_i}$ $\oplus TS_{Usr_i} \oplus TS_S)$

FIGURE 2. **Summary of login and authentication phases in Rana *et al.*'s scheme.**

Finally, SC_{Usr_i} sends the message $AuthReq = \{\overline{DID_{Usr_i}}, \omega_{Usr_i}, \theta_{Usr_i}, TS_{Usr_i}\}$ to the server S via open channel.

C. AUTHENTICATION PHASE

The server S first receives the message $AuthReq = \{\overline{DID_{Usr_i}}, \omega_{Usr_i}, \theta_{Usr_i}, TS_{Usr_i}\}$ from the user Usr_i and proceeds with the following steps in order to establish a session key with Usr_i:

- Step $Auth_1$: The server S first validates the timestamp TS_{Usr_i} in the received message $AuthReq$ from Usr_i by the condition: $(TS^*_{Usr_i} - TS_{Usr_i}) \leq \delta TS$, where the maximum allowable transmission delay for a message is denoted by δTS and $TS^*_{Usr_i}$ is the received timestamp of the message $AuthReq$. Now, if the timestamp is valid, the login request is accepted by S; otherwise, it is rejected by S.
- Step $Auth_2$: S proceeds to extract the identity ID_{Usr_i} by computing $(ID_{Usr_i} \Vert y_{Usr_i}) = Dec_{K_s}[\overline{DID_{Usr_i}}]$. After that the following calculations are performed by the server S:

$$\begin{aligned} \alpha^*_{Usr_i} &= CHash((ID_{Usr_i} \oplus a) \Vert b), \\ y^*_{Usr_i} &= \omega_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i}) \\ &\quad \oplus CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus TS_{Usr_i}), \\ \theta^*_{Usr_i} &= CHash(ID_{Usr_i} \Vert \alpha^*_{Usr_i} \Vert y^*_{Usr_i} \Vert \\ &\quad (\alpha^*_{Usr_i} \oplus y^*_{Usr_i}) \Vert TS_{Usr_i}). \end{aligned}$$

After that, S checks the legitimacy of the validating condition: $\theta^*_{Usr_i} = \theta_{Usr_i}$. If it is valid, S proceeds to the next step; otherwise, the request is rejected.

User (Usr_i)	Smart Card (SC_{Usr_i})
Input ID_{Usr_i}, current password $Pwd^o_{Usr_i}$ and new password $Pwd^n_{Usr_i}$	
	Compute $m_{Usr_i} = \eta_{Usr_i} \oplus CHash(ID_{Usr_i} \| \| Pwd^o_{Usr_i})$, $RPW^o_{Usr_i} = CHash(m_{Usr_i} \| \| Pwd^o_{Usr_i})$, $\alpha^o_{Usr_i} = \beta_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus RPW^o_{Usr_i})$, $y^o_{Usr_i} = \gamma_{Usr_i} \oplus CHash(\alpha^o_{Usr_i} \oplus RPW^o_{Usr_i})$, $\psi^o_{Usr_i} = CHash(ID_{Usr_i} \| \| RPW^o_{Usr_i}$ $\| \| y^o_{Usr_i} \| \| \alpha^o_{Usr_i})$ Check if $\psi^o_{Usr_i} = \psi_{Usr_i}$? If so, accept password change request Calculate the following: $RPW^n_{Usr_i} = CHash(m_{Usr_i} \| \| Pwd^n_{Usr_i})$, $\beta^n_{Usr_i} = \alpha^o_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus RPW^n_{Usr_i})$, $\gamma^n_{Usr_i} = y^o_{Usr_i} \oplus CHash(\alpha^o_{Usr_i} \oplus RPW^n_{Usr_i})$, $\psi^n_{Usr_i} = CHash(ID_{Usr_i} \| \| RPW^n_{Usr_i} \| \| y^o_{Usr_i} \| \| \alpha^o_{Usr_i})$, $\eta^n_{Usr_i} = m_{Usr_i} \oplus CHash(ID_{Usr_i} \| \| Pwd^n_{Usr_i})$ Update $\{\beta_{Usr_i}, \gamma_{Usr_i}, \psi_{Usr_i}, \eta_{Usr_i}\}$ by $\{\beta^n_{Usr_i}, \gamma^n_{Usr_i}, \psi^n_{Usr_i}, \eta^n_{Usr_i}\}$ in its memory

FIGURE 3. Summary of password change phase in Rana *et al.*'s scheme.

- Step $Auth_3$: S then generates a fresh timestamp TS_S and calculates the following parameter:

$$\mu_{Usr_i} = CHash(ID_{Usr_i}||y^*_{Usr_i}||(\alpha^*_{Usr_i} \oplus y^*_{Usr_i})||TS_S).$$

  Now, S sends the authentication response message $AuthRes = \{\mu_{Usr_i}, TS_S\}$ to the user Usr_i via open channel.
- Step $Auth_4$: The validity of the timestamp TS_S is checked by the condition: $(TS^*_S - TS_S) \leq \delta TS$, once the message $AuthRes = \{\mu_{Usr_i}, TS_S\}$ is received by Usr_i at time TS^*_S. If the timestamp is valid, Usr_i calculates $\mu^*_{Usr_i} = CHash(ID_{Usr_i} ||y^*_{Usr_i}|| (\alpha^*_{Usr_i} \oplus y^*_{Usr_i}) ||TS_S)$ and verifies if $\mu^*_{Usr_i} = \mu_{Usr_i}$ or not. If the validation is passed, Usr_i computes the session key shared with the server S as $SK = CHash(ID^*_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus y^*_{Usr_i} \oplus TS_{Usr_i} \oplus TS_S)$. Similarly, the server S also calculates the same session key shared with Usr_i as $SK = CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus y^*_{Usr_i} \oplus TS_{Usr_i} \oplus TS_S)$.

The login and authentication phases are briefed in Figure 2.

D. PASSWORD CHANGE PHASE

Suppose a legal registered user, say Usr_i wants to update his/her credential (password) due to security reasons. For this goal, a user authentication protocol should allow Usr_i to update his/her credentials at any time and locally without contacting the server S. The following involved steps are given below:

- Step $PwdC_1$: Usr_i inputs his/her identity ID_{Usr_i}, current password $Pwd^o_{Usr_i}$ and new password $Pwd^n_{Usr_i}$. The smart card SC_{Usr_i} of Usr_i then calculates $m_{Usr_i} = \eta_{Usr_i} \oplus CHash(ID_{Usr_i} ||Pwd^o_{Usr_i})$, $RPW^o_{Usr_i} = CHash(m_{Usr_i} ||Pwd^o_{Usr_i})$, $\alpha^o_{Usr_i} = \beta_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus RPW^o_{Usr_i})$, $y^o_{Usr_i} = \gamma_{Usr_i} \oplus CHash(\alpha^o_{Usr_i} \oplus RPW^o_{Usr_i})$ and $\psi^o_{Usr_i} = CHash(ID_{Usr_i} ||RPW^o_{Usr_i} ||y^o_{Usr_i} ||\alpha^o_{Usr_i})$. If $\psi^o_{Usr_i} = \psi_{Usr_i}$, SC_{Usr_i} accepts the password change request of the user Usr_i; else, the request is rejected.
- Step $PwdC_2$: Now, SC_{Usr_i} calculates the following with respect to new password $Pwd^n_{Usr_i}$:

$$\begin{aligned} RPW^n_{Usr_i} &= CHash(m_{Usr_i}||Pwd^n_{Usr_i}), \\ \beta^n_{Usr_i} &= \alpha^o_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus RPW^n_{Usr_i}), \\ \gamma^n_{Usr_i} &= y^o_{Usr_i} \oplus CHash(\alpha^o_{Usr_i} \oplus RPW^n_{Usr_i}), \\ \psi^n_{Usr_i} &= CHash(ID_{Usr_i}||RPW^n_{Usr_i}||y^o_{Usr_i} \\ &\quad ||\alpha^o_{Usr_i}), \\ \eta^n_{Usr_i} &= m_{Usr_i} \oplus CHash(ID_{Usr_i}||Pwd^n_{Usr_i}). \end{aligned}$$

- Step $PwdC_3$: Finally, $\{\beta_{Usr_i}, \gamma_{Usr_i}, \psi_{Usr_i}, \eta_{Usr_i}\}$ are updated with $\{\beta^n_{Usr_i}, \gamma^n_{Usr_i}, \psi^n_{Usr_i}, \eta^n_{Usr_i}\}$ in the smart card SC_{Usr_i}.

This phase is also summarized in Figure 3.

IV. CRYPTANALYSIS OF RANA *et al.*'s SCHEME

This section shows the following serious security pitfalls that are found in Rana *et al.*'s scheme [23]. We utilize the attack model that is described in Section II for cryptanalysis of Rana *et al.*'s scheme.

A. STOLEN SMART CARD AND PRIVILEGED-INSIDER ATTACKS

The stolen smart card and privileged-insider attacks are not new attacks, rather they are very well-known attacks [6], [27], [28]. In practice, the registration is done through secure channel usually by submitting the documents to a registration authority. Hence, in most cases, the registration takes place via offline mode. Due to this reason, there is a high possibility to know the registration details (documents/information) submitted by the registered users to the trusted registration authority. However, an insider user of the registration authority, being a privileged-insider attacker, has an opportunity to capture the registration details submitted by the users during the registration time.

Note that both the DY and CK-adversary models will only allow an adversary to compromise the communication channels along with the session states and private keys through the session-hijacking attacks, during the communication only. However, the stolen smart attack and privileged-insider attack require physical capture of smart card of a valid registered user. Hence, there is no connection of the DY and CK adversarial models during registration phase.

Though the server S is treated as a trusted entity in the network, but a privileged-insider user of that server S, may act as an insider attacker, say $\mathcal{AE}$, [6], [27], [28]. $\mathcal{AE}$ performs stolen smart card and privileged-insider attacks as follows.

- Step 1. Suppose during a legal user Usr_i's registration phase, $\mathcal{AE}$ knows the registration information $\{ID_{Usr_i}, RPW_{Usr_i}\}$, where $RPW_{Usr_i} = CHash(m_{Usr_i} || Pwd_{Usr_i})$. Furthermore, assume that after registration process, $\mathcal{AE}$ attains Usr_i's smart card $SC_{Usr_i} = \{\beta_{Usr_i}, \gamma_{Usr_i}, \overline{DID_{Usr_i}}, \psi_{Usr_i}, \eta_{Usr_i}, CHash(\cdot)\}$, and extracts all the credentials stored in SC_{Usr_i} using the power analysis attacks [26].
- Step 2. With the help of the credentials $\{ID_{Usr_i}, RPW_{Usr_i}\}$ and $\{\beta_{Usr_i}, \gamma_{Usr_i}, \overline{DID_{Usr_i}}, \psi_{Usr_i}, \eta_{Usr_i}\}$, $\mathcal{AE}$ computes the following:

$$\begin{aligned}\alpha_{Usr_i} &= \beta_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus RPW_{Usr_i}),\\ y_{Usr_i} &= \gamma_{Usr_i} \oplus CHash(\alpha_{Usr_i} \oplus RPW_{Usr_i}),\\ \psi^*_{Usr_i} &= CHash(ID_{Usr_i} || RPW_{Usr_i} || y_{Usr_i}\\ &\quad || \alpha_{Usr_i}),\end{aligned}$$

  and checks if $\psi^*_{Usr_i} = \psi_{Usr_i}$. If it is valid, the next step is executed.
- Step 3. $\mathcal{AE}$ guesses a password, say Pwd^* for the user Usr_i, and calculates

$$\begin{aligned}m^*_{Usr_i} &= \eta_{Usr_i} \oplus CHash(ID_{Usr_i} || Pwd^*),\\ RPW^*_{Usr_i} &= CHash(m^*_{Usr_i} || Pwd^*).\end{aligned}$$

  Now, $\mathcal{AE}$ checks the legitimacy of the condition: $RPW^*_{Usr_i} = RPW_{Usr_i}$. If it holds, $\mathcal{AE}$ is successful in guessing the user Usr_i's correct password, that is $Pwd_{Usr_i} = Pwd^*$. Otherwise, $\mathcal{AE}$ continues to guess another password and continues from Step 3.

It is then clear that $\mathcal{AE}$ can guess the correct password Pwd_{Usr_i} and obtain sensitive secret credentials $\{\alpha_{Usr_i}, y_{Usr_i}\}$ for the user Usr_i, using stolen smart card and privileged-insider attacks.

B. USER IMPERSONATION ATTACK

In this attack scenario, we again assume that a privileged-insider user of the server S will act as an insider attacker, say $\mathcal{AE}$, who knows the registration information $\{ID_{Usr_i}, RPW_{Usr_i}\}$ of a valid registered user Usr_i. Moreover, assume that $\mathcal{AE}$ has temporary access to the smart card SC_{Usr_i} of the user Usr_i, obtains all the credentials $\{\beta_{Usr_i}, \gamma_{Usr_i}, \overline{DID_{Usr_i}}, \psi_{Usr_i}, \eta_{Usr_i}\}$ stored in SC_{Usr_i} using the power analysis attacks [26] and computes the sensitive secret credentials $\{\alpha_{Usr_i}, y_{Usr_i}\}$ as discussed in Section IV-A. In addition, $\mathcal{AE}$ can also intercept the messages $AuthReq = \{\overline{DID_{Usr_i}}, \omega_{Usr_i}, \theta_{Usr_i}, TS_{Usr_i}\}$ and $AuthRes = \{\mu_{Usr_i}, TS_S\}$ during the login and authentication phases exchanges between Usr_i and S in the earlier session.

The user impersonation attack executed by $\mathcal{AE}$ is as follows:

- Step 1. On behalf of the user Usr_i, the attacker $\mathcal{AE}$ generates a fresh timestamp $TS^f_{Usr_i}$ for calculating

$$\begin{aligned}\omega^f_{Usr_i} &= y_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus \alpha_{Usr_i})\\ &\quad \oplus CHash(ID_{Usr_i} \oplus \alpha_{Usr_i} \oplus TS^f_{Usr_i}),\\ \theta^f_{Usr_i} &= CHash(ID_{Usr_i} || \alpha_{Usr_i} || y_{Usr_i} ||\\ &\quad (\alpha_{Usr_i} \oplus y_{Usr_i}) || TS^f_{Usr_i}).\end{aligned}$$

  $\mathcal{AE}$ then sends the message $AuthReq^f = \{\overline{DID_{Usr_i}}, \omega^f_{Usr_i}, \theta^f_{Usr_i}, TS^f_{Usr_i}\}$ to the server S via open channel, using the intercepted $\overline{DID_{Usr_i}}$.
- Step 2. The server S the validates the timestamp $TS^f_{Usr_i}$ in the received message $AuthReq^f$. Since the timestamp is valid, the login request is accepted by S and S extracts the identity ID_{Usr_i} by computing $(ID_{Usr_i} || y_{Usr_i}) = Dec_{K_s}[\overline{DID_{Usr_i}}]$. S also computes

$$\begin{aligned}\alpha^*_{Usr_i} &= CHash((ID_{Usr_i} \oplus a) || b),\\ y^*_{Usr_i} &= \omega^f_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i})\\ &\quad \oplus CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus TS^f_{Usr_i}),\\ \theta^*_{Usr_i} &= CHash(ID_{Usr_i} || \alpha^*_{Usr_i} || y^*_{Usr_i} ||\\ &\quad (\alpha^*_{Usr_i} \oplus y^*_{Usr_i}) || TS^f_{Usr_i}).\end{aligned}$$

  After that, S checks if $\theta^*_{Usr_i} = \theta^f_{Usr_i}$. Since this condition will also pass, S will generate a fresh timestamp TS_S and calculate the following parameter:

$$\mu_{Usr_i} = CHash(ID_{Usr_i} || y^*_{Usr_i} || (\alpha^*_{Usr_i} \oplus y^*_{Usr_i}) || TS_S).$$

  Next, S sends the authentication response message $AuthRes = \{\mu_{Usr_i}, TS_S\}$ towards the user Usr_i via open channel.
- Step 3. The adversary $\mathcal{AE}$ intercepts and blocks the message $AuthRes = \{\mu_{Usr_i}, TS_S\}$. $\mathcal{AE}$ now checks the validity of timestamp TS_S. Since the timestamp validation passes, $\mathcal{AE}$ calculates $\mu^f_{Usr_i} = CHash(ID_{Usr_i} || y_{Usr_i} || (\alpha_{Usr_i} \oplus y_{Usr_i}) || TS_S)$ and verifies if $\mu^f_{Usr_i} = \mu_{Usr_i}$ or not. Since the validation is successful, $\mathcal{AE}$ computes the session key shared with the server S as $SK = CHash(ID_{Usr_i} \oplus \alpha_{Usr_i} \oplus y_{Usr_i} \oplus TS^f_{Usr_i} \oplus TS_S)$.

User (Usr_i)	Adversary ($\mathcal{AE}$)	Server (S)
$\vdots$ $AuthReq = \{\overline{DID_{Usr_i}}, \omega_{Usr_i}, \theta_{Usr_i}, TS_{Usr_i}\}$ $\longrightarrow$	$\vdots$	$\vdots$
$\vdots$	$\vdots$ Generate fresh timestamp $TS^f_{Usr_i}$ Calculate $\omega^f_{Usr_i} = y_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus \alpha_{Usr_i})$ $\oplus CHash(ID_{Usr_i} \oplus \alpha_{Usr_i} \oplus TS^f_{Usr_i})$, $\theta^f_{Usr_i} = CHash(ID_{Usr_i} \| \| \alpha_{Usr_i} \| \| y_{Usr_i} \| \|$ $(\alpha_{Usr_i} \oplus y_{Usr_i}) \| \| TS^f_{Usr_i})$ $AuthReq^f = \{\overline{DID_{Usr_i}}, \omega^f_{Usr_i}, \theta^f_{Usr_i}, TS^f_{Usr_i}\}$ $\longrightarrow$	$\vdots$
		Check validity of timestamp If so, extract $(ID_{Usr_i} \| \| y_{Usr_i})$ $= Dec_{K_s}[\overline{DID_{Usr_i}}]$ Calculate $\alpha^*_{Usr_i} = CHash((ID_{Usr_i} \oplus a) \| \| b)$, $y^*_{Usr_i} = \omega^f_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i})$ $\oplus CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus \boxed{TS^f_{Usr_i}})$, $\theta^*_{Usr_i} = CHash(ID_{Usr_i} \| \| \alpha^*_{Usr_i} \| \| y^*_{Usr_i} \| \|$ $(\alpha^*_{Usr_i} \oplus y^*_{Usr_i}) \| \| \boxed{TS^f_{Usr_i}})$ Check if $\boxed{\theta^*_{Usr_i} = \theta^f_{Usr_i}}$? If so, generate fresh timestamp TS_S Compute $\mu_{Usr_i} = CHash(ID_{Usr_i} \| \| y_{Usr_i} \| \|$ $(\alpha^*_{Usr_i} \oplus y^*_{Usr_i}) \| \| TS_S)$ $AuthRes = \{\mu_{Usr_i}, TS_S\}$ $\longleftarrow$
	Intercept and block message $AuthRes$ Check validity of timestamp If valid, calculate $\mu^f_{Usr_i} = CHash(ID_{Usr_i} \| \| y_{Usr_i} \| \|$ $(\alpha_{Usr_i} \oplus y_{Usr_i}) \| \| TS_S)$ Verify if $\mu^f_{Usr_i} = \mu_{Usr_i}$? If so, compute session key shared with S as $SK = CHash(ID_{Usr_i} \oplus \alpha_{Usr_i} \oplus y_{Usr_i}$ $\oplus TS^f_{Usr_i} \oplus TS_S)$	Compute session key shared with $\mathcal{AE}$ as $SK = CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus y^*_{Usr_i}$ $\oplus TS^f_{Usr_i} \oplus TS_S)$

FIGURE 4. Illustration of user impersonation attack in Rana *et al.*'s scheme.

It is then evident from the above discussion that $\mathcal{AE}$ can easily perform user impersonation attack on behalf of a legal registered user Usr_i. This attack scenario is also depicted in Figure 4.

Remark 1: By simply eavesdropping the messages $AuthReq = \{\overline{DID_{Usr_i}}, \omega_{Usr_i}, \theta_{Usr_i}, TS^{(j)}_{Usr_i}\}$ *and* $AuthRes = \{\mu_{Usr_i}, TS^{(j)}_S\}$ *during the login and authentication phases in Rana et al.'s scheme for* j^{th} *session* ($j = 1, 2, 3, \ldots$)*, the adversary* $\mathcal{AE}$ *having the credentials* $\{\alpha_{Usr_i}, y_{Usr_i}, ID_{Usr_i}\}$*, can always compute the session key in the* j^{th} *session as* $SK = CHash(ID_{Usr_i} \oplus \alpha_{Usr_i} \oplus y_{Usr_i} \oplus TS^{(j)}_{Usr_i} \oplus TS^{(j)}_S)$. *As a result, Rana et al.'s scheme fails to provide forward and backward secrecy.*

C. PASSWORD CHANGE ATTACK

As discussed in Section IV-A, an adversary $\mathcal{AE}$ can guess the correct password Pwd_{Usr_i} and obtain sensitive secret credentials $\{\alpha_{Usr_i}, y_{Usr_i}\}$ for a registered authorized user Usr_i, through the stolen smart card and privileged-insider attacks. In the following, we now show that $\mathcal{AE}$ can also update his/her own password in the stolen smart card SC_{Usr_i} of the user Usr_i by involving the following steps:

- Step 1. $\mathcal{AE}$ inputs identity ID_{Usr_i}, guessed correct password $Pwd^g_{Usr_i}$ and his/her own new chosen password $Pwd^f_{\mathcal{AE}}$. The smart card SC_{Usr_i} of Usr_i then calculates $m_{Usr_i} = \eta_{Usr_i} \oplus CHash(ID_{Usr_i} \| \| Pwd^g_{Usr_i})$, $RPW^o_{Usr_i} = CHash(m_{Usr_i} \| \| Pwd^g_{Usr_i})$, $\alpha^o_{Usr_i} = \beta_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus RPW^g_{Usr_i})$, $y^o_{Usr_i} = \gamma_{Usr_i} \oplus CHash(\alpha^o_{Usr_i} \oplus RPW^o_{Usr_i})$ and $\psi^o_{Usr_i} = CHash(ID_{Usr_i} \| \| RPW^o_{Usr_i} \| \| y^o_{Usr_i} \| \| \alpha^o_{Usr_i})$. If $\psi^o_{Usr_i} = \psi_{Usr_i}$, SC_{Usr_i} accepts the password change request of the user Usr_i; else, the request is rejected.
- Step 2. SC_{Usr_i} calculates the following with respect to new password $Pwd^f_{\mathcal{AE}}$:

$$RPW^f_{Usr_i} = CHash(m_{Usr_i} \| \| Pwd^f_{\mathcal{AE}}),$$
$$\beta^n_{Usr_i} = \alpha^o_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus RPW^f_{Usr_i}),$$

Attacker ($\mathcal{AE}$)	Smart Card (SC_{Usr_i})
Input ID_{Usr_i}, correct guessed password $Pwd^g_{Usr_i}$ and new chosen password $Pwd^f_{\mathcal{AE}}$	
	Compute $m_{Usr_i} = \eta_{Usr_i} \oplus CHash(ID_{Usr_i} \| \| Pwd^g_{Usr_i})$, $RPW^o_{Usr_i} = CHash(m_{Usr_i} \| \| Pwd^g_{Usr_i})$, $\alpha^o_{Usr_i} = \beta_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus RPW^o_{Usr_i})$, $y^o_{Usr_i} = \gamma_{Usr_i} \oplus CHash(\alpha^o_{Usr_i} \oplus RPW^o_{Usr_i})$, $\psi^o_{Usr_i} = CHash(ID_{Usr_i} \| \| RPW^o_{Usr_i} \| \| y^o_{Usr_i} \| \| \alpha^o_{Usr_i})$ Check if $\psi^o_{Usr_i} = \psi_{Usr_i}$? If so, accept password change request Calculate the following: $RPW^f_{Usr_i} = CHash(m_{Usr_i} \| \| Pwd^f_{\mathcal{AE}})$, $\beta^n_{Usr_i} = \alpha^o_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus RPW^f_{Usr_i})$, $\gamma^n_{Usr_i} = y^o_{Usr_i} \oplus CHash(\alpha^o_{Usr_i} \oplus RPW^f_{Usr_i})$, $\psi^n_{Usr_i} = CHash(ID_{Usr_i} \| \| RPW^f_{Usr_i} \| \| y^o_{Usr_i} \| \| \alpha^o_{Usr_i})$, $\eta^n_{Usr_i} = m_{Usr_i} \oplus CHash(ID_{Usr_i} \| \| Pwd^f_{\mathcal{AE}})$ Update $\{\beta_{Usr_i}, \gamma_{Usr_i}, \psi_{Usr_i}, \eta_{Usr_i}\}$ by $\{\beta^n_{Usr_i}, \gamma^n_{Usr_i}, \psi^n_{Usr_i}, \eta^n_{Usr_i}\}$ in its memory

FIGURE 5. Illustration of password change attack in Rana *et al.*'s scheme.

$$\gamma^n_{Usr_i} = y^o_{Usr_i} \oplus CHash(\alpha^o_{Usr_i} \oplus RPW^f_{Usr_i}),$$
$$\psi^n_{Usr_i} = CHash(ID_{Usr_i}||RPW^f_{Usr_i}||y^o_{Usr_i} ||\alpha^o_{Usr_i}),$$
$$\eta^n_{Usr_i} = m_{Usr_i} \oplus CHash(ID_{Usr_i}||Pwd^f_{\mathcal{AE}}).$$

Finally, $\{\beta_{Usr_i}, \gamma_{Usr_i}, \psi_{Usr_i}, \eta_{Usr_i}\}$ are now updated with $\{\beta^n_{Usr_i}, \gamma^n_{Usr_i}, \psi^n_{Usr_i}, \eta^n_{Usr_i}\}$ in the smart card SC_{Usr_i}.

Hence, it is clear from the discussion that $\mathcal{AE}$ can easily update Usr_i's password with a newly chosen fake password and use the smart card SC_{Usr_i} for accessing the service in future communications in Rana *et al.*'s scheme. This attack scenario is depicted in Figure 5.

D. EPHEMERAL SECRET LEAKAGE (ESL) ATTACK

According to the attack model discussed in Section II, in order to provide ESL attack protection against an adversary $\mathcal{AE}$ under the CK-adversary model [25], a session key between two entities should be based on temporal (short-term) secrets (for example, random secrets) as well as long-term (permanent) secrets (for example, long-term secrets, private keys, etc.). However, in Rana *et al.*'s scheme, the session key between a legal user Usr_i and the server S is created as $SK = CHash(ID_{Usr_i} \oplus \alpha_{Usr_i} \oplus y_{Usr_i} \oplus TS_{Usr_i} \oplus TS_S)$, where the timestamps TS_{Usr_i} and TS_S are generated by Usr_i and S, respectively. Based on the discussion in Remark 1, since $\mathcal{AE}$ has the credentials $\{\alpha_{Usr_i}, y_{Usr_i}, ID_{Usr_i}\}$, he/she can easily calculate the session keys SK in any session. Moreover, each session key SK does not include any random secrets (temporal secrets). Thus, Rana *et al.*'s scheme does not protect ESL attack under the CK-adversary model.

E. LACK OF UNTRACEABILITY

In this section, we show that Rana *et al.*'s scheme fails to provide untraceability property, which is also illustrated in Figure 6. Assume that an adversary $\mathcal{AE}$ intercepts the authentication request messages during login and authentication phases between a registered user Usr_i and the server S in two sessions, namely j^{th} and l^{th} sessions. It is worth to notice that $\overline{DID_{Usr_i}}$ remains static in both sessions, where $\overline{DID_{Usr_i}} = Enc_{K_S}[ID_{Usr_i} \; ||y_{Usr_i}]$. During the login and authentication phase of Rana *et al.*'s scheme, the server S only sends the message $AuthRes = \{\mu_{Usr_i}, TS_S\}$ and not any dynamic $\overline{DID_{Usr_i}}$. As a result, $\overline{DID_{Usr_i}}$ remains static over successive sessions only. This is another design flaw that is existed in Rana *et al.*'s scheme too. This clearly proves that if the same user Usr_i interacts with the server over j^{th} and l^{th} sessions, it is detected by $\mathcal{AE}$.

F. USELESS PARAMETERS CALCULATION

During the login and authentication phases (see Figure 2), the server S extracts the identity ID_{Usr_i} by computing $(ID_{Usr_i} \; ||y_{Usr_i}) = Dec_{K_s}[\overline{DID_{Usr_i}}]$. After that the following calculations are performed by the server S:

$$\alpha^*_{Usr_i} = CHash((ID_{Usr_i} \oplus a)||b),$$
$$\boxed{y^*_{Usr_i}} = \omega_{Usr_i} \oplus CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i}) \oplus CHash(ID_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus TS_{Usr_i}),$$
$$\theta^*_{Usr_i} = CHash(ID_{Usr_i}||\alpha^*_{Usr_i}||y^*_{Usr_i}|| (\alpha^*_{Usr_i} \oplus y^*_{Usr_i})||TS_{Usr_i}).$$

It is clear that, even without computing $y^*_{Usr_i}$, the server S can still compute $\theta^*_{Usr_i}$ with the help of the decrypted y_{Usr_i}

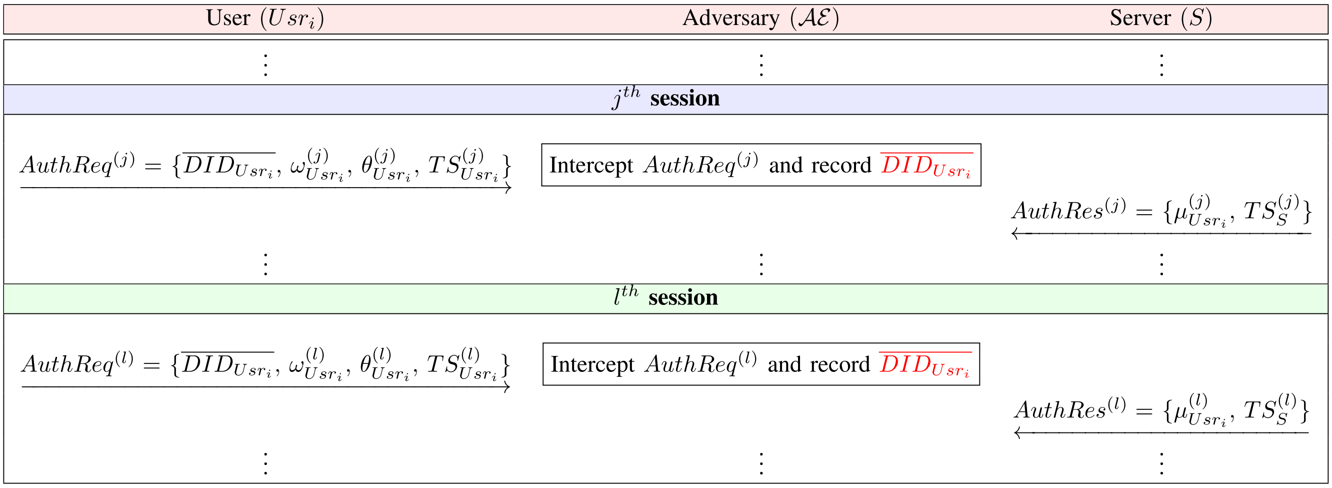

FIGURE 6. **Illustration of untraceability in Rana *et al.*'s scheme.**

from $Dec_{K_s}[\overline{DID_{Usr_i}}]$ in order to check $\theta^*_{Usr_i} = \theta_{Usr_i}$. Thus, it is unnecessary to calculate the parameter $y^*_{Usr_i}$.

V. POSSIBLE REMEDIES

We provide some possible remedies that can overcome the security weaknesses found in the analyzed Rana *et al.*'s scheme [23]. We apply the user biometrics as third factor to improve the security in Rana *et al.*'s scheme. A fuzzy extractor is a popular biometrics verification technique [29], which is composed of the following two functions:

- $Gen(\cdot)$: It takes a user's biometric BIO_{Usr_i} as input and gives a biometric secret key σ_{Usr_i} of l_b bits, say and another public reproduction parameter τ_{Usr_i}, that is, $Gen(BIO_{Usr_i} = (\sigma_{Usr_i}, \tau_{Usr_i})$. This function is randomize or probabilistic in nature.
- $Rep(\cdot)$: It takes a noisy user's biometric BIO'_{Usr_i} and public reproduction parameter τ_{Usr_i}, and results the original biometric secret key σ_{Usr_i}, that is, $Rep(BIO'_{Usr_i}, \tau_{Usr_i})$ under the restriction that the Hamming distance between original BIO_{Usr_i} and noisy BIO'_{Usr_i} is less than or equal to a predefined threshold value.

Remedy #1. *Protection against privileged-insider and stolen smart card attacks*

We provide the following modifications in Rana *et al.*'s scheme to protect against privileged-insider and stolen smart card attacks:

1) During the registration phase, the user Usr_i can additionally pick another random secret r_{Usr_i} and also a temporary identity TID_{Usr_i} and calculate $RPW_{Usr_i} = CHash(m_{Usr_i} || Pwd_{Usr_i})$ and $RPW'_{Usr_i} = RPW_{Usr_i} \oplus r_{Usr_i}$. Next, Usr_i needs to send the registration request as $RReq = \{TID_{Usr_i}, ID_{Usr_i}, RPW'_{Usr_i}\}$ to the server S via a secure channel.
2) After reception of $RReq$ from the user Usr_i, S picks a random secret y_{Usr_i} for Usr_i and its own two secrets a and b for computing the following components:
$$\overline{DID_{Usr_i}} = Enc_{K_S}[ID_{Usr_i} || y_{Usr_i}],$$
$$\alpha_{Usr_i} = CHash((ID_{Usr_i} \oplus a) || b),$$
$$\beta'_{Usr_i} = \alpha_{Usr_i} \oplus RPW'_{Usr_i},$$
$$\gamma'_{Usr_i} = y_{Usr_i} \oplus RPW'_{Usr_i},$$
$$\psi'_{Usr_i} = CHash(ID_{Usr_i} || y_{Usr_i} || \alpha_{Usr_i})$$
S then inserts the information $\{TID_{Usr_i}, \beta'_{Usr_i}, \gamma'_{Usr_i}, \psi'_{Usr_i}, CHash(\cdot)\}$ into a smart card SC_{Usr_i} and sends the registration response $RRes = SC_{Usr_i}$ to Usr_i via secure channel. S stores $(TID_{Usr_i}, \overline{DID_{Usr_i}})$ in its secure database.
3) After receiving $RRes$, Usr_i imprints his/her personal biometrics BIO_{Usr_i} to compute $Gen(BIO_{Usr_i} = (\sigma_{Usr_i}, \tau_{Usr_i})$. After that Usr_i calculates $\eta_{Usr_i} = m_{Usr_i} \oplus CHash(ID_{Usr_i} || Pwd_{Usr_i} || \sigma_{Usr_i})$ and inserts it into SC_{Usr_i}. Furthermore, Usr_i calculates
$$\begin{aligned}\beta_{Usr_i} &= (\beta'_{Usr_i} \oplus RPW'_{Usr_i}) \\ &\quad \oplus CHash(ID_{Usr_i} || \sigma_{Usr_i} || RPW_{Usr_i}) \\ &= \alpha_{Usr_i} \oplus CHash(ID_{Usr_i} || \sigma_{Usr_i} \\ &\quad || RPW_{Usr_i}), \\ \gamma_{Usr_i} &= (\gamma'_{Usr_i} \oplus RPW'_{Usr_i}) \\ &\quad \oplus CHash(ID_{Usr_i} || \sigma_{Usr_i} || \alpha_{Usr_i} \\ &\quad || RPW_{Usr_i}) \\ &= y_{Usr_i} \oplus CHash(ID_{Usr_i} || \sigma_{Usr_i} \\ &\quad \alpha_{Usr_i} || RPW_{Usr_i}), \\ \psi_{Usr_i} &= CHash(\psi'_{Usr_i} || RPW_{Usr_i} || \sigma_{Usr_i}) \\ &= CHash(CHash(ID_{Usr_i} || y_{Usr_i} || \\ &\quad \alpha_{Usr_i}) || RPW_{Usr_i} || \sigma_{Usr_i}).\end{aligned}$$
Usr_i then updates $\{\beta'_{Usr_i}, \gamma'_{Usr_i}, \psi'_{Usr_i}\}$ with $\{\beta_{Usr_i}, \gamma_{Usr_i}, \psi_{Usr_i}\}$. Thus, SC_{Usr_i} has the credentials $\{TID_{Usr_i}, \beta_{Usr_i}, \gamma_{Usr_i}, \psi_{Usr_i}, \eta_{Usr_i}, CHash(\cdot)\}$.

It is clear that an adversary $\mathcal{AE}$ only knows ID_{Usr_i}, but does not have knowledge of Pwd_{Usr_i}, σ_{Usr_i} and m_{Usr_i}. Thus, both privileged-insider and stolen smart card attacks will not be succeeded by the adversary $\mathcal{AE}$.

Remedy #2. *Protection against user impersonation attacks*

The following modifications in Rana *et al.*'s scheme are needed to protect against user impersonation attack and as a consequence, an ESL attack too:

1) After inserting the smart card SC_{Usr_i}, the user Usr_i inputs his/her credentials, like the identity $ID^*_{Usr_i}$ and password $Pwd^*_{Usr_i}$. Usr_i also imprints his/her biometrics, say BIO'_{Usr_i} and calculates $Rep(BIO'_{Usr_i}, \tau_{Usr_i}) = \sigma_{Usr_i}$. Then, SC_{Usr_i} calculates the following:

$$\begin{aligned} m_{Usr_i} &= \eta_{Usr_i} \oplus CHash(ID^*_{Usr_i}||Pwd^*_{Usr_i} \\ &\quad ||\sigma_{Usr_i}), \\ RPW^*_{Usr_i} &= CHash(m_{Usr_i}||Pwd^*_{Usr_i}), \\ \alpha^*_{Usr_i} &= \beta_{Usr_i} \oplus CHash(ID_{Usr_i}||\sigma_{Usr_i} \\ &\quad ||RPW_{Usr_i}), \\ y^*_{Usr_i} &= \gamma_{Usr_i} \oplus CHash(ID_{Usr_i}||\sigma_{Usr_i} \\ &\quad \alpha_{Usr_i}||RPW_{Usr_i}), \\ \psi^*_{Usr_i} &= CHash(CHash(ID_{Usr_i}||y_{Usr_i}|| \\ &\quad \alpha_{Usr_i})||RPW_{Usr_i}||\sigma_{Usr_i}). \end{aligned}$$

Next, SC_{Usr_i} checks the validity of $\psi^*_{Usr_i} = \psi_{Usr_i}$. If it holds, the login request of Usr_i is accepted by the smart card SC_{Usr_i}. Otherwise, the phase is terminated here.

2) SC_{Usr_i} calculates the following components by generating a fresh timestamp TS_{Usr_i} and a fresh random secret r_1:

$$\begin{aligned} \omega_{Usr_i} &= y^*_{Usr_i} \oplus CHash(ID^*_{Usr_i} \oplus \alpha^*_{Usr_i}) \\ &\quad \oplus CHash(ID^*_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus TS_{Usr_i}), \\ r^*_1 &= CHash(r_1||TS_{Usr_i}) \oplus CHash(ID^*_{Usr_i} \\ &\quad ||y^*_{Usr_i}||TS_{Usr_i}||\alpha^*_{Usr_i}) \\ \theta_{Usr_i} &= CHash(ID^*_{Usr_i}||\alpha^*_{Usr_i}||y^*_{Usr_i} \\ &\quad ||CHash(r_1||TS_{Usr_i})|| \\ &\quad (\alpha^*_{Usr_i} \oplus y^*_{Usr_i})||TS_{Usr_i}). \end{aligned}$$

Finally, SC_{Usr_i} sends the message $AuthReq = \{TID_{Usr_i}, \omega_{Usr_i}, r^*_1, \theta_{Usr_i}, TS_{Usr_i}\}$ to the server S via open channel.

3) After receiving the message $AuthReq$, the server S validates timestamp TS_{Usr_i}. If the timeliness if valid, S fetches $\overline{DID_{Usr_i}}$ corresponding to TID_{Usr_i} from its secure database. Additionally, S extracts the identity ID_{Usr_i} and permanent secret y_{Usr_i} by computing $(ID_{Usr_i} ||y_{Usr_i}) = Dec_{K_s}[\overline{DID_{Usr_i}}]$. After that the following calculations are executed by the server S:

$$\begin{aligned} \alpha^*_{Usr_i} &= CHash((ID_{Usr_i} \oplus a)||b), \\ r'_1 &= r^*_1 \oplus CHash(ID_{Usr_i} \\ &\quad ||y_{Usr_i}||TS_{Usr_i}||\alpha^*_{Usr_i}) \\ \theta^*_{Usr_i} &= CHash(ID_{Usr_i}||\alpha^*_{Usr_i}||y_{Usr_i}|| \\ &\quad r'_1||(\alpha^*_{Usr_i} \oplus y_{Usr_i})||TS_{Usr_i}). \end{aligned}$$

S checks the legitimacy of $\theta^*_{Usr_i} = \theta_{Usr_i}$. If it is valid, S generates a fresh timestamp TS_S, a fresh random secret r_2 and a new temporary identity $TID^n_{Usr_i}$, and calculates $\mu_{Usr_i} = CHash(ID_{Usr_i} ||y_{Usr_i}|| (\alpha^*_{Usr_i} \oplus y^*_{Usr_i}) ||TS_S)$, $r^*_2 = CHash(r_2||TS_S) \oplus CHash(ID_{Usr_i} ||\alpha^*_{Usr_i} ||y_{Usr_i} ||TS_S)$, the session key shared with Usr_i as $SK_{S,U} = CHash(TID_{Usr_i} \oplus ID_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus y_{Usr_i} \oplus TS_{Usr_i} \oplus TS_S \oplus r'_1 \oplus CHash(r_2 ||TS_S))$, the session key verifier $SKV_{S,U} = CHash(SK_{S,U} ||TS_S)$ and $TID^*_{Usr_i} = TID^n_{Usr_i} \oplus CHash(TID_{Usr_i} ||SK_{S,U} ||TS_S)$ for sending the authentication response message $AuthRes = \{TID^*_{Usr_i}, \mu_{Usr_i}, r^*_2, SKV_{S,U}, TS_S\}$ to the user Usr_i via open channel.

4) Usr_i now checks the timeliness of TS_S. If it is valid, Usr_i computes $\mu^*_{Usr_i} = CHash(ID_{Usr_i} ||y^*_{Usr_i}|| (\alpha^*_{Usr_i} \oplus y^*_{Usr_i}) ||TS_S)$ and verifies if $\mu^*_{Usr_i} = \mu_{Usr_i}$ or not. If the validation is passed, Usr_i computes $r'_2 = r^*_2 \oplus CHash(ID_{Usr_i} ||\alpha^*_{Usr_i} ||y^*_{Usr_i} ||TS_S)$, the session key shared with the server S as $SK_{U,S} = CHash(TID_{Usr_i} \oplus ID^*_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus y^*_{Usr_i} \oplus TS_{Usr_i} \oplus TS_S \oplus CHash(r_1 ||TS_{Usr_i}) \oplus r'_2)$, the session key verifier $SKV_{U,S} = CHash(SK_{U,S} ||TS_S)$ and $TID^n_{Usr_i} = TID^*_{Usr_i} \oplus CHash(TID_{Usr_i} ||SK_{U,S} ||TS_S)$. If $SKV_{U,S} = SKV_{S,U}$, the session key validation passes and Usr_i updates TID_{Usr_i} with new $TID^n_{Usr_i}$ in the smart card SC_{Usr_i}.

Thus, at the end of this phase both Usr_i and S are sharing the same session key $SK_{U,S} = CHash(TID_{Usr_i} \oplus ID^*_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus y^*_{Usr_i} \oplus TS_{Usr_i} \oplus TS_S \oplus CHash(r_1 ||TS_{Usr_i}) \oplus r'_2) = CHash(TID_{Usr_i} \oplus ID_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus y_{Usr_i} \oplus TS_{Usr_i} \oplus TS_S \oplus r'_1 \oplus CHash(r_2 ||TS_S)) = CHash(TID_{Usr_i} \oplus ID_{Usr_i} \oplus \alpha^*_{Usr_i} \oplus y_{Usr_i} \oplus TS_{Usr_i} \oplus TS_S \oplus CHash(r_1 ||TS_{Usr_i}) \oplus CHash(r_2 ||TS_S))$ $(= SK_{S,U})$. It is worth noticing that the session key relies on both the permanent (long-term) secrets (ID_{Usr_i}, α_{Usr_i} and y_{Usr_i}) which cannot be now derived through stolen smart card and privileged-insider attacks, and temporal (short-term) secrets (r_1 and r_2). Hence, under the CK-adversary model, an adversary $\mathcal{AE}$ requires to know both the temporal and permanent secrets in order to compromise the session keys in different sessions between Usr_i and S. As a result, ESL attack is protected in our proposed remedy. Additionally, forward and backward secrecy goals are also preserved in this remedy.

Remedy #3. *Untraceability preservation*

From the discussion provided in our Remedy #2, instead of sending static $\overline{DID_{Usr_i}}$, temporary identity TID_{Usr_i} is sent in the message $AuthReq = \{TID_{Usr_i}, \omega_{Usr_i}, r^*_1, \theta_{Usr_i}, TS_{Usr_i}\}$ and it is again updated with new random identity $TID^n_{Usr_i}$ by the user Usr_i after verifying the message $AuthRes = \{TID^*_{Usr_i}, \mu_{Usr_i}, r^*_2, SKV_{S,U}, TS_S\}$, where $TID^*_{Usr_i} = TID^n_{Usr_i} \oplus CHash(TID_{Usr_i} ||SK_{S,U} ||TS_S)$. The adversary $\mathcal{AE}$ cannot link the messages during a particular session with other subsequent sessions between the user Usr_i and the server S, because all the components in the messages are dynamic and unique due to utilization of random secrets and timestamps. Thus, in our remedy, it is clear that untraceability and anonymity are safeguarded.

Remedy #4. *Protection against user password change attack*

In this remedy, we show that a legal registered user Usr_i can update his/her credentials at any time and locally without contacting the server S. The following involved steps are given below.

1) Usr_i inputs his/her identity ID_{Usr_i}, current password $Pwd^o_{Usr_i}$, and imprints current biometrics $BIO^o_{Usr_i}$. The smart card SC_{Usr_i} of Usr_i calculates $Rep(BIO^o_{Usr_i}, \tau_{Usr_i}) = \sigma^o_{Usr_i}$, along with

$$\begin{aligned} m^o_{Usr_i} &= \eta_{Usr_i} \oplus CHash(ID_{Usr_i}||Pwd^o_{Usr_i} \\ &\quad ||\sigma^o_{Usr_i}), \\ RPW^o_{Usr_i} &= CHash(m_{Usr_i}||Pwd^o_{Usr_i}), \\ \alpha^o_{Usr_i} &= \beta_{Usr_i} \oplus CHash(ID_{Usr_i}||\sigma^o_{Usr_i} \\ &\quad ||RPW^o_{Usr_i}), \\ y^o_{Usr_i} &= \gamma_{Usr_i} \oplus CHash(ID_{Usr_i}||\sigma^o_{Usr_i} \\ &\quad \alpha^o_{Usr_i}||RPW^o_{Usr_i}), \\ \psi^o_{Usr_i} &= CHash(CHash(ID_{Usr_i}||y^o_{Usr_i}|| \\ &\quad \alpha^o_{Usr_i})||RPW^o_{Usr_i}||\sigma^o_{Usr_i})). \end{aligned}$$

Next, SC_{Usr_i} checks the validity of $\psi^o_{Usr_i} = \psi_{Usr_i}$. If it holds, the user password and biometrics change request is accepted by the smart card SC_{Usr_i}. Otherwise, the phase is terminated here.

2) SC_{Usr_i} prompts the user Usr_i to input new password $Pwd^n_{Usr_i}$ and imprint new biometrics $BIO^n_{Usr_i}$. SC_{Usr_i} then computes $Gen(BIO^n_{Usr_i} = (\sigma^n_{Usr_i}, \tau^n_{Usr_i}), \eta^n_{Usr_i} = m^o_{Usr_i} \oplus CHash(ID_{Usr_i} ||Pwd^n_{Usr_i} ||\sigma^n_{Usr_i})$ along with the following parameters:

$$\begin{aligned} \beta^n_{Usr_i} &= \alpha^o_{Usr_i} \oplus CHash(ID_{Usr_i}||\sigma^n_{Usr_i} \\ &\quad ||RPW^n_{Usr_i}), \\ \gamma^n_{Usr_i} &= y^o_{Usr_i} \oplus CHash(ID_{Usr_i}||\sigma^n_{Usr_i} \\ &\quad \alpha^o_{Usr_i}||RPW^n_{Usr_i}), \\ \psi^n_{Usr_i} &= CHash(CHash(ID_{Usr_i}||y^o_{Usr_i}|| \\ &\quad \alpha^o_{Usr_i})||RPW^n_{Usr_i}||\sigma^n_{Usr_i}). \end{aligned}$$

Finally, $\{\beta_{Usr_i}, \gamma_{Usr_i}, \psi_{Usr_i}, \eta_{Usr_i}, \tau_{Usr_i}\}$ are updated with $\{\beta^n_{Usr_i}, \gamma^n_{Usr_i}, \psi^n_{Usr_i}, \eta^n_{Usr_i}, \tau^n_{Usr_i}\}$ in the smart card SC_{Usr_i}.

VI. CONCLUSION

This comment paper reviewed a recently proposed Rana *et al.*'s scheme and pointed out several security weaknesses like stolen smart card attack, privileged-insider attack, user impersonation attack, password change attack and ESL attack. Moreover, their scheme fails to provide untraceability feature. We applied the fuzzy extractor method for biometrics verification to provide more security of the system. To remedy the security pitfalls in Rana *et al.*'s scheme, we provided four remedies that successfully overcome the security weaknesses found in Rana *et al.*'s scheme. Thus, we significantly improved the security of Rana *et al.*'s scheme in this comment paper.

REFERENCES

[1] N. Chen and M. Okada, "Towards 6G Internet of Things and the convergence with RoF system," *IEEE Internet Things J.*, early access, Dec. 28, 2020, doi: 10.1109/JIOT.2020.3047613.

[2] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for Internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[3] S. Chatterjee, A. K. Das, and J. K. Sing, "An enhanced access control scheme in wireless sensor networks," *Ad Hoc Sensor Wireless Netw.*, vol. 21, nos. 1–2, pp. 121–149, Jan. 2014.

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

[5] M. S. Obaidat, S. P. Rana, T. Maitra, D. Giri, and S. Dutta, *Biometric Security and Internet of Things (IoT)*. Cham, Switzerland: Springer, 2019, pp. 477–509.

[6] A. K. Das, A. K. Sutrala, S. Kumari, V. Odelu, M. Wazid, and X. Li, "An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 2070–2092, Sep. 2016.

[7] C. Lin, D. He, N. Kumar, K.-K.-R. Choo, A. Vinel, and X. Huang, "Security and privacy for the Internet of drones: Challenges and solutions," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 64–69, Jan. 2018.

[8] C. T. Li, C. C. Lee, and C. Y. Weng, "Security and efficiency enhancement of robust ID based mutual authentication and key agreement scheme preserving user anonymity in mobile networks," *J. Inf. Sci. Eng.*, vol. 34, no. 1, pp. 155–170, Jan. 2018.

[9] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K.-K.-R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9390–9401, Sep. 2020.

[10] C.-T. Li, C.-L. Chen, C.-C. Lee, C.-Y. Weng, and C.-M. Chen, "A novel three-party password-based authenticated key exchange protocol with user anonymity based on chaotic maps," *Soft Comput.*, vol. 22, no. 8, pp. 2495–2506, Apr. 2018.

[11] Y. Zhang, D. He, L. Li, and B. Chen, "A lightweight authentication and key agreement scheme for Internet of drones," *Comput. Commun.*, vol. 154, pp. 455–464, Mar. 2020.

[12] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981.

[13] N.-Y. Lee and Y.-C. Chiu, "Improved remote authentication scheme with smart card," *Comput. Standards Interfaces*, vol. 27, no. 2, pp. 177–180, Jan. 2005.

[14] E.-J. Yoon, E.-K. Ryu, and K.-Y. Yoo, "An improvement of Hwang–Lee–Tang's simple remote user authentication scheme," *Comput. Secur.*, vol. 24, no. 1, pp. 50–56, Feb. 2005.

[15] M. K. Khan, J. Zhang, and X. Wang, "Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices," *Chaos, Solitons Fractals*, vol. 35, no. 3, pp. 519–524, Feb. 2008.

[16] L. Fan, J.-H. Li, and H.-W. Zhu, "An enhancement of timestamp-based password authentication scheme," *Comput. Secur.*, vol. 21, no. 7, pp. 665–667, Nov. 2002.

[17] J.-J. Shen, C.-W. Lin, and M.-S. Hwang, "Security enhancement for the timestamp-based password authentication scheme using smart cards," *Comput. Secur.*, vol. 22, no. 7, pp. 591–595, Oct. 2003.

[18] M. S. Hwang and C.-Y. Liu, "Authenticated encryption schemes: Current status and key issues," *Int. J. Netw. Secur.*, vol. 1, no. 2, pp. 61–73, 2005.

[19] A. Irshad, H. Naqvi, S. A. Chaudhary, M. Usman, M. Shafiq, O. Mir, and A. Kanwal, "Cryptanalysis and improvement of a multi-server authenticated key agreement by Chen and Lee's scheme," *Inf. Technol. Control*, vol. 47, no. 3, pp. 431–446, 2018.

[20] C.-T. Li and M.-S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *J. Netw. Comput. Appl.*, vol. 33, pp. 1–5, Jan. 2010.

[21] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 824–839, Sep. 2018.

[22] S. D. Kaul and A. K. Awasthi, "Security enhancement of an improved remote user authentication scheme with key agreement," *Wireless Pers. Commun.*, vol. 89, no. 2, pp. 621–637, Jul. 2016.
[23] M. Rana, A. Shafiq, I. Altaf, M. Alazab, K. Mahmood, S. A. Chaudhry, and Y. B. Zikria, "A secure and lightweight authentication scheme for next generation IoT infrastructure," *Comput. Commun.*, vol. 165, pp. 85–96, Jan. 2021.
[24] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.
[25] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, Amsterdam, The Netherlands, 2002, pp. 337–351.
[26] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.
[27] M. Wazid, A. K. Das, V. K. Bhat, and A. V. Vasilakos, "LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment," *J. Netw. Comput. Appl.*, vol. 150, Jan. 2020, Art. no. 102496.
[28] A. Vangala, A. K. Das, and J. Lee, "Provably secure signature-based anonymous user authentication protocol in an Internet of Things-enabled intelligent precision agricultural environment," *Concurrency Comput., Pract. Exper.*, p. e6187, Mar. 2021, doi: 10.1002/cpe.6187.
[29] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, Jan. 2008.

ASHOK KUMAR DAS (Senior Member, IEEE) received the M.Tech. degree in computer science and data processing, the M.Sc. degree in mathematics from IIT Kharagpur, India, and the Ph.D. degree in computer science and engineering. He is currently an Associate Professor with the Center for Security, Theory, and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. He has authored over 260 articles in international journals and conferences in the above areas, including over 220 reputed journal articles. Some of his research findings are published in top cited journals, such as the IEEE TRANSACTIONS ON SMART GRID, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS (formerly IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE), *Computers & Electrical Engineering*, *Computer Networks*, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE ACCESS, IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, *IEEE Communications Magazine*, *IEEE Consumer Electronics Magazine*, *Computer Methods and Programs in Biomedicine*, *Future Generation Computer Systems*, *Expert Systems with Applications*, *Computer Standards & Interfaces*, and the *Journal of Network and Computer Applications*. His research interests include cryptography, network security, blockchain, security in the Internet of Things (IoT), Internet of Vehicles (IoV), Internet of Drones (IoD), smart grids, smart city, cloud/fog computing, intrusion detection, and AI/ML security. He has served as a Program Committee Member in many international conferences. He also severed as one of the Technical Program Committee Chairs of the first International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, in June 2019, and the second International Congress on Blockchain and Applications (BLOCKCHAIN'20), L'Aquila, Italy, in October 2020. He is on the editorial board of IEEE SYSTEMS JOURNAL, the *Journal of Networks and Computer Applications* (Elsevier), *Computer Communications* (Elsevier), *IET Communications*, *KSII Transactions on Internet and Information Systems*, and the *International Journal of Internet Technology and Secured Transactions* (Inderscience). He is a Guest Editor of *Computers & Electrical Engineering* (Elsevier) for the special issue on Big data and IoT in e-healthcare, *Wireless Communications and Mobile Computing* (Wiley-Hindawi) for the special issue on Security and Privacy for Smart Mobile Devices: Attacks, Challenges, and New Designs, and *ICT Express* (Elsevier) for the special issue on Blockchain Technologies and Applications for 5G Enabled IoT.

BASUDEB BERA received the M.Sc. degree in mathematics and computing from IIT (ISM), Dhanbad, India, in 2014, and the M.Tech. degree in computer science and data processing from IIT Kharagpur, India, in 2017. He is currently pursuing the Ph.D. degree in computer science and engineering with the Center for Security, Theory, and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. He has published 15 articles in international journals and conferences in his research areas. His research interests include cryptography, network security, and blockchain technology.

MOHAMMAD WAZID (Senior Member, IEEE) received the M.Tech. degree in computer network engineering from Graphic Era University, Dehradun, India, and the Ph.D. degree in computer science and engineering from the International Institute of Information Technology, Hyderabad, India. He was also a Postdoctoral Researcher with the Cyber Security and Networks Laboratory, Innopolis University, Innopolis, Russia. He is currently working as an Associate Professor with the Department of Computer Science and Engineering, and the Head of the Cyber Security and IoT Research Group, Graphic Era University. He has published over 85 research articles in international journals and conferences. His current research interests include security, remote user authentication, the Internet of Things (IoT), cloud computing, blockchain, and AI/ML security. He received the University Gold Medal in his master's degree and also awarded as the Young Scientist by USSTC (UCOST), Department of Science and Technology, Government of Uttarakhand, India.

SAJJAD SHAUKAT JAMAL received the Ph.D. degree in mathematics from Quaid-i-Azam University, Islamabad, Pakistan. He is currently working as an Assistant Professor with the Department of Mathematics, King Khalid University, Abha, Saudi Arabia. He has several journal articles in his research areas. His research interests include number theory, cryptography, digital watermarking, steganography, information security, multimedia security, and blockchain technology.

YOUNGHO PARK (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Kyungpook National University, Daegu, South Korea, in 1989, 1991, and 1995, respectively. From 1996 to 2008, he was a Professor with the School of Electronics and Electrical Engineering, Sangju National University, South Korea. From 2003 to 2004, he was a Visiting Scholar with the School of Electrical Engineering and Computer Science, Oregon State University, USA. He is currently a Professor with the School of Electronics Engineering, Kyungpook National University. His research interests include information security, computer networks, and multimedia.