IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

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

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

SEMS-5G: A Secure and Efficient Multi-Server Authentication Scheme for 5G Networks

Azeem Irshad[1] , Mohammed Alreshoodi [2,*]

[1]Department of computer science and software engineering, International Islamic University Islamabad, Pakistan
[2] Unit of Scientific Research, Applied College, Qassim University, Buraydah, Saudi Arabia (email: mo.alreshoodi@qu.edu.sa)

*Corresponding authors: (e-mail: mo.alreshoodi@qu.edu.sa)

ABSTRACT. The fifth-generation (5G) network is regarded as a key enabler technology for promoting the Internet of Things (IoT) and overcoming the corresponding challenges in the future, such as the support of low communication latency, high data rates, and managing numerous connections to devices in IoT-based ecosystem. To meet such requirements with the realization of 5G network technology as well as the qualification for cloud-based services, the resource deficient mobile end users must gain secure access to remote cloud computing servers. A robust multiserver authentication may ensure the stipulated computational efficiency for authenticated key agreements in 5G networks. Many Multi-Server Authentication (MSA) protocols have been presented so far for various applications. Yet, the compliance to perfect forward secrecy (PFS), untraceability, and privacy-based security features, along with the resilience to de-synchronization and other known attacks, is uncertain. Recently, Wu *et al*. presented another MSA scheme for a distributed cloud-based 5G environment. Although the scheme fulfills PFS; however, we identified that Wu *et al*. is prone to impersonation attack, password guessing attack, and man-in-the-middle attack. We have demonstrated an efficient and secure multiserver authentication protocol SEMS-5G ensuring PFS and all other significant security properties that previous schemes could not offer. The results of SEMS-5G are validated using automated ProVerif tool and formally analyzed using BAN logic analysis. The analysis and results prove that our scheme supports all security features at an economical cost.

INDEX TERMS 5G network, Internet of Things, Cloud computing, Multi-server authentication, Perfect forward secrecy

I. INTRODUCTION

The 5G network technology has proved to be a key propellant in realizing the future demands of the internet of things (IoT)-based ecosystem. The researchers have streamlined their focus in this particular area of interest recently. The emerging applications of IoT call for defining new performance standards in the spheres related to IoT, such as artificial intelligence, big data, managing innumerable network connections, latency, power requirements, network coverage, and security, etc. The IoT network brings a world of tiny objects in contact with each other, while those objects may encompass smart homes, health, transportation, industry, and military fields. A substantial fragment of the information is derived from IoT-based sensors. It is also projected that by the end of the year 2020, the growth curve of IoT devices may approach the figure of fifty billion [1, 2]. Besides, the cultural shift of priorities of end users in favor of mobile gadgets not only leads to the exponential growth of IoT devices but also uncovers an arena of openings for malicious adversaries. Without appropriate security measures, it would be untenable to induct the IoT-based applications and exchange sensitive nature of data. The sensors deployed in the fields may collect and transmit data to other devices or intermediate gateways. In a few applications, the user may also directly access the sensor devices. The bulk of the data produced by IoT sensors is accessed and stored on cloud servers, which is later available to authenticated end users. The data capturing and transmission to end users may involve diverse network domains, including field sensors, data centers, cloud servers, edge computing nodes, and mobile end users [3-5].

The heterogeneity of 5G networks, as defined above, calls for an efficient and secure protocol framework for multiple servers ensuring privacy, perfect forward secrecy, and resistance to known attacks across the diverse network domains. The online services delivered by distributed cloud servers in 5G technology must be protected from unauthorized access at each level of a heterogeneous network domain [6-8]. In general, the IoT environment is exposed to many risks that could lead to sudden power breakdown and network disruptions. The sensitive nature of captured data needs to be communicated using encryption keys (session key) established out of mutually agreed and secure authenticated key agreements. Due to the power deficient devices as well as distributed and diverse domain-based architecture, there is a need to design an efficient yet secure multiserver authentication protocol for 5G networks. Although there are many multiserver authentication protocols that are presented before to address the requirements of various applications. However, most of those schemes are employing costly public key cryptosystem-based operations, which render those protocols inapplicable for a power deficient environment [27, 37]. In this study, we review a MSA-based scheme by Wu *et al.* presented for a distributed cloud-based 5G environment [35]. The scheme provides many useful security features; however, we discover that Wu *et al.* is still prone to impersonation attack, man-in-the-middle attack, and password guessing attacks. With this consideration, we propose a novel multiserver authenticated key agreement scheme which is efficient due to the use of low cost symmetric operations, as well as secure since it adheres to deliver PFS, untraceability, privacy and resistance to known attacks.

A. RELATED WORK

In 1981, the Lamport [9] presented a pioneer password-based authentication protocol over a public channel. Nonetheless, the main drawback of this protocol was to consult a password table with lots of other attacks and overheads. Then, there were a few improved protocols [10-12] to cover the drawbacks in [9]. In 2001, Chang and Wu [13] and Hwang *et al*. [14] presented authentication protocols based on smart cards. Later, many other smart card-based authenticated key agreements were presented [15-18]. Li *et al*. [19] put forward the identity authentication scheme employing the neural networks for multiserver environment. In addition, to enhance the security of login the factor of the smart card was complemented with biometric factor and was termed as three-factor authentication. Lately, many researchers presented three-factor authentication schemes to boost the security of protocols [20]. The multiserver architecture is being adopted for some time to aid the authentication procedures of resource deficient devices by gaining computational efficiencies.

Following the same multiserver paradigm, many security solutions have been presented [21-29]. Wu *et al*. [30] designed a multiserver authentication protocol for distributed cloud-based architecture; however, their scheme was defenseless against a stolen device attack as well as a privileged insider attack. Afterward, Wu *et al.* [35] found that the scheme [30] was prone to privileged insider threats and did not fulfill perfect forward secrecy [35]. Also, [35] presented an improved scheme; however, it was found as vulnerable to Man-In-The-Middle (MIDM) attack as well as identity and password guessing threats. Later, Amin and Biswas [33] designed a lightweight authenticated key agreement protocol for IoT-based devices in distributed cloud architecture. Nonetheless, the scheme was suffering from privacy problems and was prone to offline-password guessing attack. The Tsai and Lo [27] presented another authentication protocol for a distributed cloud environment; however, the protocol was prone to many attacks. After that, Irshad et al. [29] presented a multiserver authentication protocol employing bilinear pairing operations for a distributed mobile cloud environment; however, it was found to be susceptible to many problems [35]. Then, Mollah et al. [37] identified few security limitations in edge computing and demonstrated the comparative analysis on privacy problems of edge computing. Onwards, Kunal et al. [36] also discussed different security applications of fog and edge computing. Then, Irshad et al. [46] designed an anonymous multiserver authentication protocol enabling the construction of session key from offline registration centre with the use of Elliptic Curve Cryptography (ECC) operations. The scheme, however, employed costly ECC operations in comparison with other symmetric key-based schemes. Later, Ying et al. [44] presented a lightweight user authentication protocol for multi-server 5G networks employing self-certified public key cryptography, however, the scheme was susceptible to user impersonation, password guessing and stolen verifier threats. Thereafter, Xiong et al. [45] came forward with an efficient privacy-aware key agreement scheme having hierarchical access control for the cloud computing paradigm. However, the scheme was vulnerable to user impersonation and stolen verifier threats. Lately, Xie et al. [48] presented a PUF-enabled lightweight three-factor access control technique for multiserver paradigm, however, the scheme seems to employ PUF function without appropriate fuzzy extractor function that might lead the scheme to desynchronization attack.

Thus despite many demonstrated authentication schemes so far, there is still a need for more efficient yet secure security solutions for being implemented in a distributed cloud or 5G network environment. The summary of the related work is also presented in Table I.

B. NETWORK MODEL

The contributed model introduces a control authority to administer various cloud computing servers, as shown in Fig. 1. Cloud computing provides bulk management and storage facility of data. The multiserver authentication scenario in 5G architecture comprises three participating entities, 1) the mobile user (Ui), 2) cloud server (Sj), and 3) controlling authority (CA). The mobile device of the user supports mmWave as well as device-to-device (D2D) technologies to communicate not only with one another but also with the server. The mobile and smart devices are deficient in computational power, in general. The control authority administers many cloud servers. The cloud servers provide services to mobile end users. The control authority is responsible for registering the mobile users and assists in mutual authentication between the cloud server as well as mobile users. The contributed model provides a way of sharing the same session key among all three participants.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381616

Table I. Tabular depiction of literature work

Scheme	*Features*	*Drawbacks*	*Year*
Li et al. [19]	A multiserver authentication scheme for architecture using neural networks	Password guessing attack	2001
Wu et al. [30]	An IoT-based authentication protocol for distributed cloud computing	Lacking forward secrecy and resistence to previleged insider attack	2018
Wu et al. [35]	A smart-card-based remote user authentication scheme for multiserver environment	Prone to MIDM and password guessing attack	2020
Amin and Biswas [33]	A lightweight user authentication scheme for multi-gateway based WSN	Prone to offline password guessing threat	2016
Tsai and Lo. [27]	A privacy-aware authentication protocol for distributed mobile cloud computing environment	Prone to impersonation and forgery attacks	2015
Irshad et al. [29]	An lightweight chaotic map based authentication protocol for multiserver architecture	Prone to impersonation and forgery attacks	2016
Irshad et al. [46]	An anonymous multiserver authenticated key agreement with offline registration centre using ECC operations	Employed costly ECC operations	2018
Ying et al. [44]	Lightweight user authentication protocol for multi-server 5G networks using self-certified public key cryptography	Prone to user impersonation, password guessing and stolen verifier threats	2019
Xiong et al. [45]	An Efficient Privacy-Aware Authentication Scheme With Hierarchical Access Control for Mobile Cloud Computing Services	Prone to user impersonation and stolen verifier threats	2022
Xie et al. [48]	Physical-Unclonable-Function-Based Lightweight Three-Factor Authentication for Multiserver Architectures	Use of PUF without appropriate fuzzy extractor function might lead the scheme to desynchronization attack	2023

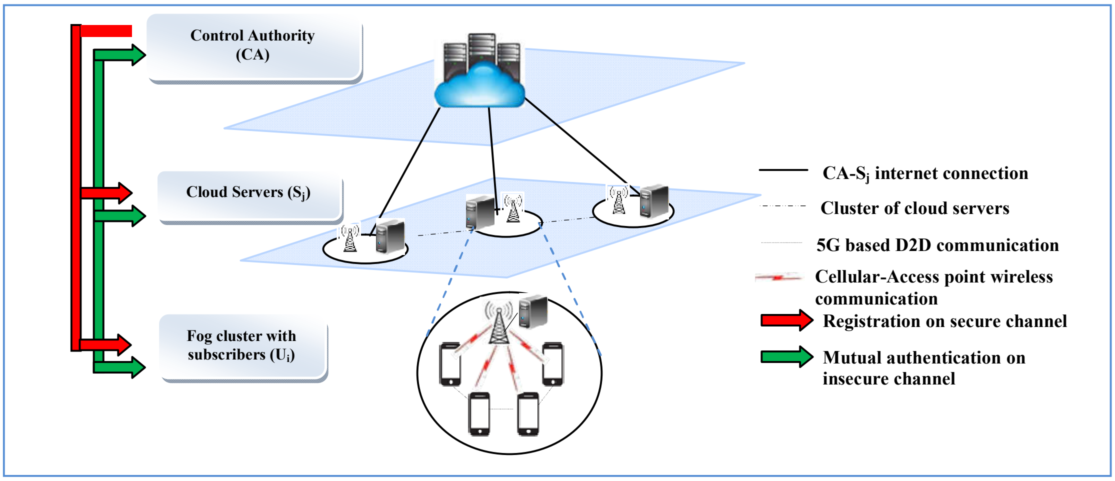

Fig. 1 Distributed cloud-based 5G architecture

C. CONTRIBUTION

The main contribution of this proposed scheme is described below:

- A lightweight and anonymous multiserver authentication protocol for the cloud-IoT-based 5G network environment has been designed.
- The informal as well as formal analysis depict that our proposed authentication scheme not only adheres to PFS, privacy and untraceability features but is also resilient from up-to-date known threats.
- A widely accepted rigorous formal security analysis ROR model has been employed to prove the security features of the contributed model.
- The performance evaluation presents the comparative analysis of various schemes with the contributed model, which affirms the strong features of the proposed scheme.

D. SCHEME ORGANIZATION

The organization of this research study is illustrated below: Section 2 presents the review and cryptanalysis of the Wu *et al.* scheme [35]. Section 3 demonstrates the proposed model. Section 4 analyzes the study on formal as well as informal grounds. Section 5 illustrates the performance evaluation analysis. The last section concludes this work.

II. REVIEW OF WU *ET AL*. SCHEME

This section presents the working and cryptanalysis of the Wu *et al*. scheme.

A. REVISITING WU ET AL. SCHEME

There are three participating entities in this protocol, i.e., the user Ui, cloud server Sj, and control authority CA. The notations used in this paper are given in Table II.

1) SERVER REGISTRATION PROCEDURE

The cloud server performs its registration process with trusted authority by following the under-mentioned steps:

Step-1: Initially, the server Sj chooses its identity SIDj and random integer es, and submits these parameters to CA.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381616

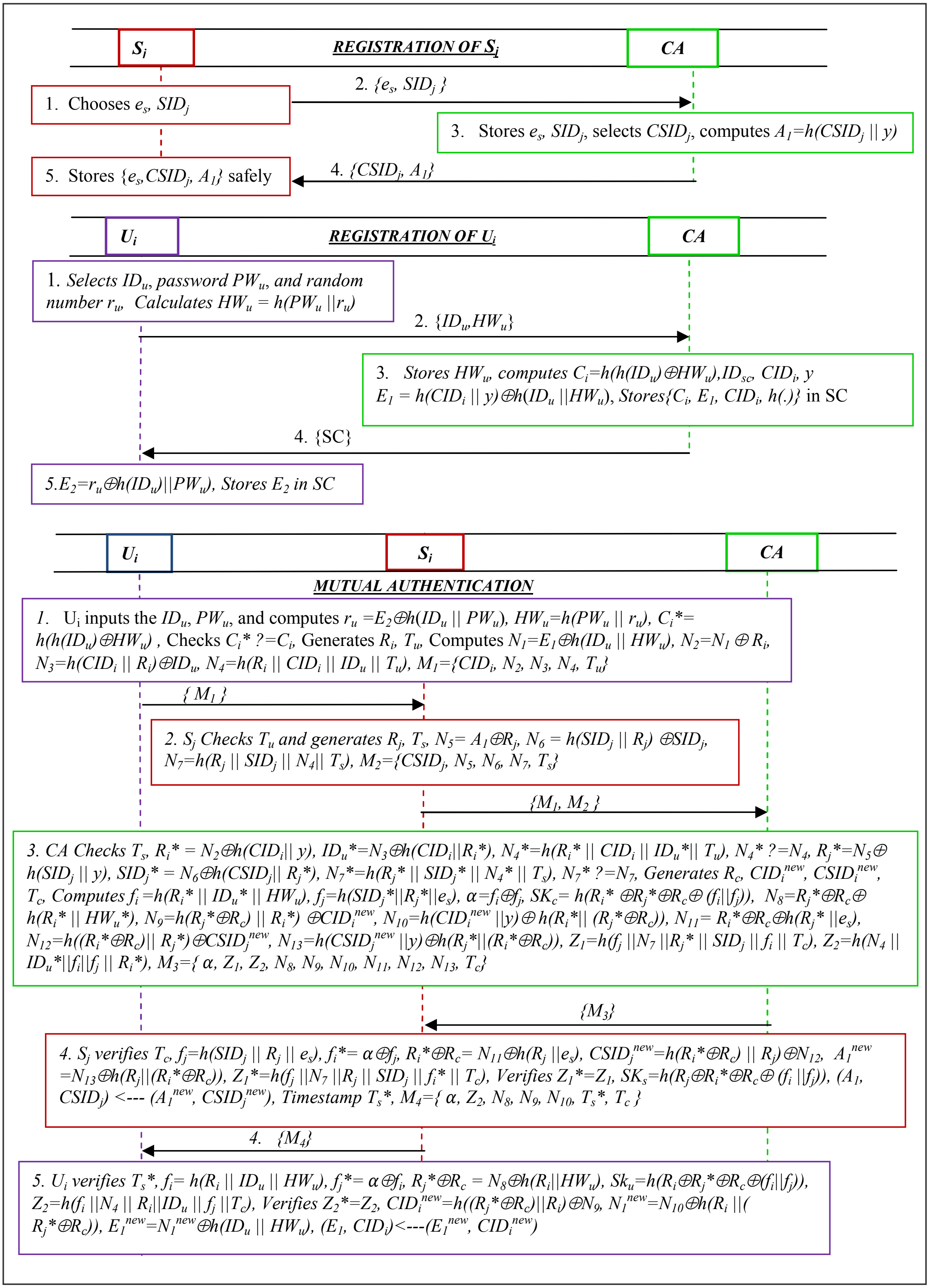

Fig. 2 Wu *et al*. scheme

Figure 3 depicts the server registration procedure pictorially.

***Step-*2**: The CA, after receiving the request, chooses pseudo-identity $CSID_j$ for server, and stores SID_j and e_s in its repository. Next, the CA computes $A_1=h(CSID_j \| y)$ and submits $\{CSID_j, A_1\}$ to Sj. The Sj stores the parameters $\{e_s, CSID_j, A_1\}$ safely.

2) USER REGISTRATION PROCEDURE

The user executes its registration process with CA by pursuing the under-mentioned procedure, as depicted in Figure 3.

***Step-*1**: The user after determining its identity ID_u, password PW_u, and selecting a random integer r_u, computes $HW_u = h(PW_u \| r_u)$, and submits the registration request $\{ID_u, PW_u\}$ to TA.

***Step-*2**: The CA stores the parameter HW_u, and calculates $C_i=h(h(ID_u)\oplus HW_u)$, generates identity of smart card ID_{sc}, pseudonym identity CID_i. Then, it further calculates $E_1 = h(CID_i \| y)\oplus h(ID_u \| HW_u)$ and stores $\{C_i, E_1, CID_i, h()\}$ in smart card (SC). Finally, it submits the SC to the user.

***Step-*3**: Next, the user calculates $E_2= r_u\oplus h(ID_u \| PW_u)$, and further adds E_2 in SC.

3) LOGIN AND AUTHENTICATION PROCEDURE

The user needs to build an agreed session key with S_j, while the *CA* assists in establishing this shared session key using the following procedure.

***Step-*1**: Initially, the U_i inputs ID_u, PW_u , and calculates $r_u =E_2\oplus h(ID_u \| PW_u)$,, $HW_u=h(PW_u \|r_u)$, $C_i^*= h(h(ID_u)\oplus HW_u)$. Then, it verifies $C_i^* ?=C_i$, and generates timestamp T_u and random integer R_i using pseudorandom number generator (PRNG). Next, it calculates $N_1=E_1\oplus h(ID_u \| HW_u)$, $N_2=N_1 \oplus R_i$, $N_3=h(CID_i \| R_i) \oplus ID_u$ and $N_4=h(R_i \| CID_i \| ID_u \| T_u)$. Next, it submits the authentication request $M_1=\{CID_i, N_2, N_3, N_4, T_u\}$ to server.

***Step-*2**: The server checks the freshness of T_u and engenders a random number R_j and timestamp T_s. Then it calculates $N_5= A_1\oplus R_j$, $N_6 = h(SID_j \| R_j) \oplus SID_j$ and $N_7=h(R_j \| SID_j \| N_4\| T_s)$. Next, it sends M_1 and $M_2=\{CSID_j, N_5, N_6, N_7, T_s\}$ to *CA*.

***Step-*3**: The CA after receiving M_1 and M_2 checks the timestamp T_s. If it is within the freshness threshold, it computes $R_i^* = N_2\oplus h(CID_i\| y)$, $ID_u^*=N_3\oplus h(CID_i\|R_i^*)$, $N_4^*=h(R_i^* \| CID_i \| ID_u^*\| T_u)$ and verifies the equality for $N_4^* ?=N_4$. If it does not match, it terminates the session, or else it calculates $R_j^*=N_5\oplus h(SID_j \| y)$, $SID_j^* = N_6\oplus h(CSID_j\| R_j^*)$, $N_7^*=h(R_j^* \| SID_j^* \| N_4^* \| T_s)$and verifies the equation for $N_7^* ?=N_7$. If it is true, then further engenders a random number R_c, pseudo-identities CID_i^{new} as well as $CSID_i^{new}$ and a fresh timestamp T_c. Then it calculates $f_i =h(R_i^*\|ID_u^*\|HW_u)$, $f_j=h(SID_j^*\|R_j^*\|e_s)$, $\alpha=f_i\oplus f_j$, $SK_c= h(R_i^* \oplus R_j^*\oplus R_c\oplus (f_i\|f_j))$, $N_8=R_j^*\oplus R_c\oplus h(R_i^* \| HW_u^*)$, $N_9=h(R_j^*\oplus R_c) \| R_i^*) \oplus CID_i^{new}$, $N_{10}=h(CID_i^{new} \|y) \oplus h(R_i^*\|(R_j^*\oplus R_c))$, $N_{11}=R_i^*\oplus R_c\oplus h(R_j^* \|e_s)$, $N_{12}=h((R_i^*\oplus R_c)\| R_j^*)\oplus CSID_j^{new}$, $N_{13}=h(CSID_j^{new} \|y) \oplus h(R_j^*\|(R_i^*\oplus R_c))$, $Z_1=h(h(CSID_j^{new} \|y)\| CSID_j^{new} \|f_j \|N_7 \|R_j^* \| SID_j \| f_i \| T_c)$ and $Z_2=h(h(CID_i^{new} \|y) \|CID_i^{new} \|N_4 \| ID_u^*\|f_i \|f_j \| R_i^*)$. Finally, it sends the message $M_3=\{ \alpha, Z_1, Z_2, N_8, N_9, N_{10}, N_{11}, N_{12}, N_{13}, T_c\}$ to S_j.

***Step-*4**: The S_j verifies T_c, and computes $f_j =h(SID_j \| R_j \| e_s)$, $f_i^*= \alpha \oplus f_j$, $R_i^*\oplus R_c= N_{11}\oplus h(R_j \|e_s)$, $CSID_j^{new}=h(R_i^*\oplus R_c) \| R_j) \oplus N_{12}$, $A_1^{new} = N_{13}\oplus h(R_j\|(R_i^*\oplus R_c))$, $Z_1^*=h(A_1^{new} \|CSID_j^{new} \|f_j\|N_7\|R_j\|SID_j\|f_i^*\| T_c)$, and verifies $Z_1^* ?=Z_1$. If it is true, it further calculates $SK_s=h(R_i^*\oplus R_j^*\oplus R_c\oplus (f_i \|f_j))$ and replaces $(A_1, CSID_j)$ with $(A_1^{new}, CSID_j^{new})$. Next, it generates timestamp T_s^* and sends the message $M_4=\{ \alpha, Z_2, N_8, N_9, N_{10}, T_s^*, T_c \}$ to user as shown in Fig. 2.

***Step-*5**: The U_i verifies T_s^*, and computes $f_i= h(R_i \| ID_u \| HW_u)$, $f_j^*= \alpha \oplus f_i$, $R_j^*\oplus R_c = N_8\oplus h(R_i\|HW_u)$, $Sk_u=h(R_i\oplus R_j^*\oplus R_c\oplus (f_i \| f_j))$, $CID_i^{new}=h((R_j^*\oplus R_c)\| R_i) \oplus N_9$, $N_1^{new}=N_{10}\oplus h(R_i \|(R_j^*\oplus R_c))$, $Z_2=h(f_i \|N_4 \| R_i\|ID_u \| f_j \|T_c)$. Then it verifies the equality for $Z_2^* ?=Z_2$. If it is not true, it terminates the session. Or else it calculates $E_1^{new}=N_1^{new}\oplus h(ID_u \| HW_u)$ and replaces (E_1, CID_i) with (E_1^{new}, CID_i^{new}) in its smart card.

Table II. Description of symbols

Symbols	Semantics
CA:	Controlling Authority
U_i, S_j:	The i^{th} user, jth cloud server
ID_u, PW_u:	Identity and password of U_i
SID_j :	Identity of cloud server S_j
ID_{sc}:	Identity of smart card for U_i
CID_i:	Pseudo-identity of U_i
$CSID_j$:	Pseudo-identity of S_j
SK_u, SK_s, SK_c	Session keys constructed by U_i, S_j and CA
T_u, T_s, T_c	Timestamps assumed by U_i, S_j and CA
n_0:	A 160-bit high entropy prime integer
$h(.)$:	One-way hash operation
$\mathcal{A}$:	Malicious adversary
e_s, R_i, R_j, R_c:	Randomly generated 160-bit integers by participants
$E_k(.)$/ $D_k(.)$:	Symmetric encryption/decryption using key k
$\|$,$\oplus$:	Concatenation and Exclusive-OR functions

B. CRYPTANALYSIS OF WU ET AL.

The Wu *et al.* is ascertained to be having four vulnerabilities, including a design limitation and few attacks such as impersonation threat, man-in-the-middle attack, and identity as well as password guessing attack. The cryptanalysis of Wu *et al.* scheme is given as follows:

1) ONE DESIGN LIMITATION

The user can never match the equality for $Z_2^*= Z_2$ i.e. $Z_2=h(N_4 \| ID_u^*\|f_i \|f_j \| R_i^*)$ as constructed by CA and $Z_2^*=h(f_i \|N_4 \| R_i\|ID_u \| f_j^*\|T_c)$ as computed by user. This suggests that the user and CA can never agree to mutually agreed session key merely because of non-matching of the same computed parameters on both ends.

2) MAN-IN-THE-MIDDLE ATTACK

The user U_i and server S_j are unable to verify the legitimacy of pseudonyms $\{CID_i^{new}, CSID_i^{new}\}$ as well as $\{h(CID_i^{new}\|y)$, $h(CSID_j^{new} \|y)\}$ parameters. An adversary may alter the N_9 and N_{12} factors, i.e. $N_9=h(R_j^*\oplus R_c)\|R_i^*)\oplus CID_i^{new}$ and $N_{12}=h((R_i^*\oplus R_c)\| R_j^*)\oplus CSID_j^{new}$ of message M_3 without coming into the knowledge of U_i and S_j, respectively. This constitutes a successful man-in-the-middle attack on Wu *et al.* by a possible adversary.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381616

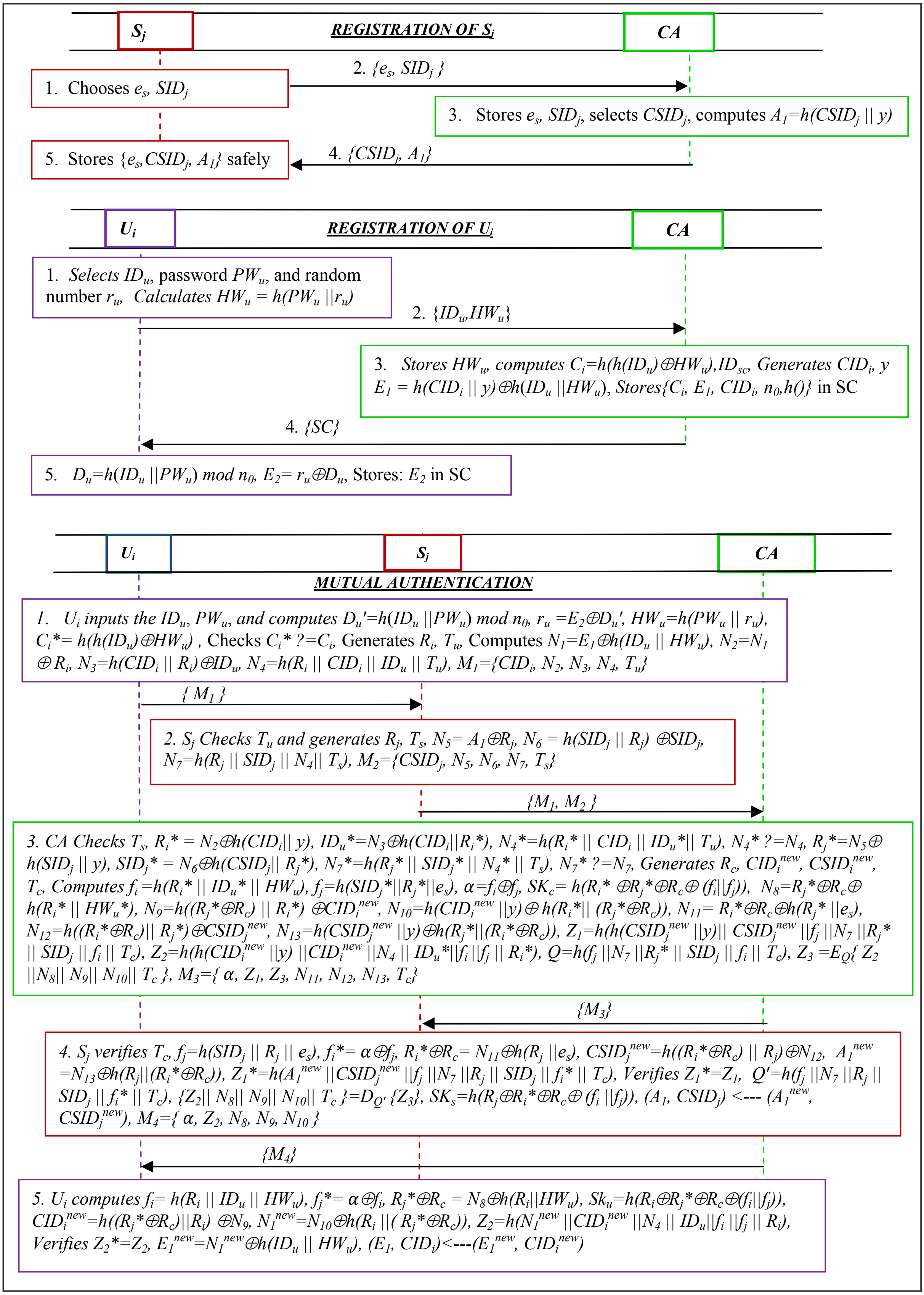

Fig. 3 Pictorial representation of SEMS-5G

```
(**************** Channels ****************)
free S_Cl:channel [private]. (*Confidential Channel*)
free P_Cl: channel. (*Insecure Channel*)
(********** Shared keys **********)
free SKu:bitstring. [private]
free SKs:bitstring. [private]
free SKc:bitstring. [private]
(********** constants and variables **********)
free y:bitstring [private]. (**The secret key of CA**)
free Ci:bitstring [private].
free es:bitstring [private].
free CIDi:bitstring [private].
free CSIDj:bitstring [private].
(*********** Constructor and Deconstructors***********)
fun h(bitstring):bitstring. (**hash digest function**)
fun addone(bitstring):bitstring. (**for addition of one**)
fun mod(bitstring, bitstring):bitstring. (**modulus**)
fun mult(bitstring, bitstring):bitstring. (**scalar multipliction**)
fun senc(bitstring, bitstring):bitstring. (**symmetric encryption**)
reduc forall z:bitstring,key:bitstring; sdec(senc(z, key),key)=z.
fun con(bitstring, bitstring):bitstring. (**concatenation**)
reduc forall m:bitstring,n:bitstring; getfirst (con(m, n))=m.
reduc forall m:bitstring,n:bitstring; getsecond(con(m, n))=n.
fun XOR(bitstring,bitstring):bitstring.
equation forall a:bitstring,b:bitstring; XOR(XOR(a,b),b)=a.
(***************** Events *****************)
event Ui_started(bitstring).
event Ui_Authed(bitstring).
(********************  Queries ********************)

query attacker(SKu).

query attacker(SKs).

query attacker(SKc).

query id:bitstring; inj-event(Ui_ Authed()) ==> inj-event (Ui_started()).
```

Fig. 4 Channels, Constants, and functions

3) IMPERSONATION ATTACK

An adversary $\mathcal{A}$, during the course of the protocol execution, may intercept and block the delivery of message on its way to the server from CA. Next, $\mathcal{A}$ may directly forward the partially selected contents of the intercepted message towards the user, while the later successfully authenticates the message, but will be unable to notice the malicious involvement of adversary circumventing the server. In this manner, the server remains ignorant of the generated session key by the participating user or CA.

4) IDENTITY AND PASSWORD GUESSING ATTACK

According to Wang *et al.* scheme [34], in Wu *et al.* the identity and password can both be guessed for being low entropy strings if not used in combination with high entropy random variables or long term secrets for computing hash digest parameters. In case an adversary steals the smart card and its contents, and then using those contents it may compute it may compute the ID_u and PW_u from E_2 by using the following steps.

Step-1: The adversary having access to E_2 and C_i calculates $r_u^* = E_3 \oplus h(ID_u^* \| PW_u^*)$, $HW_u^* = h(PW_u^* \| r_u^*)$ and $C_i' = h(h(ID_u^*) \oplus HW_u^*)$. Then it verifies the equality for $C_i' ?= C_i$.

Step-2: If this does not hold true, it selects another identity or password from the dictionary. Otherwise, it confirms the validity of the selected identity as well as the password for a particular user U_i.

III. PROPOSED MODEL (SEMS-5G)

There are three participating entities in this protocol, i.e., the user Ui, cloud server S_j, and control authority CA. Before registering the entities (users and servers), the CA selects its private key y, and a medium integer n_0 ($2^4 \leq n_0 \leq 2^8$) [35]. The proposed model comprises the registration procedures for servers and users, as well as login and authentication procedures enabling the construction of the agreed session key among the three participants.

A. SERVER REGISTRATION PROCEDURE

The cloud server completes its registration process with trusted authority by following the under-mentioned steps:

Step-1: Originally, the server S_j chooses its identity SID_j and random integer e_s, and submits these parameters to CA. Figure 3 depicts the server registration procedure pictorially.

Step-2: The CA, after receiving the request chooses pseudo-identity $CSID_j$ for server, and stores SID_j and 160-bit e_s integer in its repository. Next, CA computes $A_1 = h(CSID_j \| y)$ and submits $\{CSID_j, A_1\}$ to Sj. The Sj stores $\{e_s, CSID_j, A_1\}$ safely.

B. USER REGISTRATION PROCEDURE

The user completes its registration process with CA by pursuing the following procedure, as depicted in Figure 3.

Step-1: The user, after determining its identity ID_u, password PW_u, and selecting a random integer r_u, computes $HW_u = h(PW_u \| r_u)$, and submits the registration request $\{ID_u, PW_u\}$ to TA.

Step-2: The CA stores the parameter HW_u, and calculates $C_i = h(h(ID_u) \oplus HW_u)$, generates identity of smart card ID_{sc}, pseudonym identity CID_i. Then, it further calculates $E_1 =$

```
(***************** Ui's process*****************)
Let process_Ui=
new IDu: bitstring;
new PWu: bitstring;
new ru:bitstring;
new SIDj:bitstring;
let HWu= h(con(PWu, ru)) in
let Du=mod(h(con(IDu, PWu)), no) in
let E2=XOR(ru, Du) in
in(S_Cl,(xCi:bitstring, xE1:bitstring, xCIDi:bitstring, xno:bitstring));
!
Event Ui_started();
New PWu':bitstring;
Let let Du'=mod(h(con(IDu, PWu')), xno) in
Let ru'=XOR(E2, Du') in
Let HWu'=h(con(PWu', ru')) in
Let Ci'=h(XOR(h(IDu), Hwu')) in
if Ci' = xCi then
new Ri: bitstring;
new Tu: bitstring;
let N1=XOR(E1, h(con(IDu, PWu))) in
let N2=XOR(N1, Ri) in
let N3=XOR(h(xCIDi, Ri), IDu) in
let N4=h(con(con(Ri, xCIDi), con(IDu, Tu))) in
out(P_Cl,(xCIDi, N2, N3, N4, Tu));
in(P_Cl,(, xZ2: bitstring, xN8: bitstring, xN9: bitstring, xN10: bitstring, xTc:bitstring)
let fi=h(con(con(Ri, IDu, ), HWu) in
let fj=XOR(, fi) in
let SKu=h(XOR(XOR(XOR(xN8, h(con(Ri, HWu))), Ri), con(fi, fj))) in
let CIDinew=XOR(h(con(XOR(Rj', Rc),Ri)), xN9) in
let N1new=XOR(xN10, h(con(Ri, XOR(Rj', Rc)))) in
let Z2'=h(con(con(con(con(N1new, CIDinew), con(N4, IDu), con(fi, fj)), Ri))) in
if Z2'= xZ2 then
let E1new=XOR(N1new, h(con(IDu, HWu))) in
let E1=E1new in
let CIDi=CIDnew in
event Ui_Authed();
0).
```

Fig. 5 User Process

$h(CID_i \mid\mid y) \oplus h(ID_u \mid\mid HW_u)$ and stores $\{C_i, E_1, CID_i, n_0, h()\}$ in smart card (SC). Ultimately, it sends the SC to the user.

***Step-*3**: Next, the user computes $D_u=h(ID_u \mid\mid PW_u) \bmod n_0$, $E_2= r_u \oplus D_u$, and further adds E_2 in SC.

C. LOGIN AND AUTHENTICATION PROCEDURE

The user needs to build an agreed session key with S_j, while the *CA* assists in establishing this shared session key using the following procedure.

Step-1: Initially, the U_i inputs ID_u, PW_u , and calculates $D_u'=h(ID_u \mid\mid PW_u) \bmod n_0$, $r_u =E_2 \oplus D_u'$, $HW_u=h(PW_u \mid\mid r_u)$, $C_i^*= h(h(ID_u) \oplus HW_u)$. Then, it verifies $C_i^* \ ?{=} C_i$, and generates a random integer R_i and timestamp T_u. Next, it calculates $N_1=E_1 \oplus h(ID_u \mid\mid HW_u)$, $N_2=N_1 \oplus R_i$, $N_3=h(CID_i \mid\mid R_i) \oplus ID_u$ and $N_4=h(R_i \mid\mid CID_i \mid\mid ID_u \mid\mid T_u)$. Next, it submits the authentication request $M_1=\{CID_i, N_2, N_3, N_4, T_u\}$ to server.

Step-2: The server checks the freshness of T_u and engenders a random number R_j and timestamp T_s. Then it calculates $N_5= A_1 \oplus R_j$, $N_6 = h(SID_j \mid\mid R_j) \oplus SID_j$ and $N_7=h(R_j \mid\mid SID_j \mid\mid N_4 \mid\mid T_s)$. Next, it sends M_1 and $M_2=\{CSID_j, N_5, N_6, N_7, T_s\}$ to *CA.*

Step-3: The CA after receiving M_1 and M_2 checks the timestamp T_s. If it is within the freshness threshold, it computes $R_i^* = N_2 \oplus h(CID_i \mid\mid y)$, $ID_u^*=N_3 \oplus h(CID_i \mid\mid R_i^*)$, $N_4^*=h(R_i^* \mid\mid CID_i \mid\mid ID_u^* \mid\mid T_u)$ and verifies the equality check for $N_4^* \ ?{=} N_4$. If it is not true, it abandons the session, or else it calculates $R_j^*=N_5 \oplus h(SID_j \mid\mid y)$, $SID_j^* = N_6 \oplus h(CSID_j \mid\mid R_j^*)$, $N_7^*=h(R_j^* \mid\mid SID_j^* \mid\mid N_4^* \mid\mid T_s)$ and verifies the equality for $N_7^* \ ?{=} N_7$. If true, then further engenders a random number R_c, pseudo-identities CID_i^{new} as well as $CSID_i^{new}$ and a fresh timestamp T_c. Then it calculates $f_i =h(R_i^* \mid\mid ID_u^* \mid\mid HW_u)$, $f_j=h(SID_j^* \mid\mid R_j^* \mid\mid e_s)$, $\alpha=f_i \oplus f_j$, $SK_c= h(R_i^* \oplus R_j^* \oplus R_c \oplus (f_i \mid\mid f_j))$, $N_8=R_j^* \oplus R_c \oplus h(R_i^* \mid\mid HW_u^*)$, $N_9=h((R_j^* \oplus R_c) \mid\mid R_i^*) \oplus CID_i^{new}$, $N_{10}=h(CID_i^{new} \mid\mid y) \oplus h(R_i^* \mid\mid (R_j^* \oplus R_c))$, $N_{11}=R_i^* \oplus R_c \oplus h(R_j^* \mid\mid e_s)$, $N_{12}=h((R_i^* \oplus R_c) \mid\mid R_j^*) \oplus CSID_j^{new}$, $N_{13}=h(CSID_j^{new} \mid\mid y) \oplus h(R_j^* \mid\mid (R_i^* \oplus R_c))$, $Z_1=h(h(CSID_j^{new}$

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381616

$||y)||\ CSID_j^{new}\ ||f_j\ ||N_7\ ||R_j^*\ ||\ SID_j\ ||\ f_i\ ||\ T_c)$, $Z_2=h(h(CID_i^{new}\ ||y)\ ||CID_i^{new}\ ||N_4\ ||\ ID_u^*||f_i\ ||f_j\ ||\ R_i^*)$, $Q=h(f_j\ ||N_7\ ||R_j^*\ ||SID_j||\ f_i||T_c)$ and $Z_3=E_Q\{Z_2\ ||N_8||\ N_9||\ N_{10}||\ T_c\}$. Finally, it sends the message $M_3=\{\alpha, Z_1, Z_3, N_{11}, N_{12}, N_{13}, T_c\}$ to S_j.

***Step*-4**: The S_j verifies T_c, and computes $f_j=h(SID_j\ ||\ R_j\ ||\ e_s)$, $f_i^*=\alpha\oplus f_j$, $R_i^*\oplus R_c=N_{11}\oplus h(R_j\ ||e_s)$, $CSID_j^{new}=h((R_i^*\oplus R_c)\ ||\ R_j)\ \oplus N_{12}$, $A_1^{new}=N_{13}\oplus h(R_j||(R_i^*\oplus R_c))$, $Z_1^*=h(A_1^{new}\ ||CSID_j^{new}\ ||f_j||N_7||R_j||SID_j||f_i^*||\ T_c)$, and verifies $Z_1^*\ ?{=}Z_1$. If it is true, it further calculates $Q'=h(f_j\ ||N_7\ ||R_j\ ||\ SID_j\ ||\ f_i^*\ ||\ T_c)$, $\{Z_2||\ N_8||\ N_9||\ N_{10}||\ T_c\}=D_{Q'}\{Z_3\}$, $SK_s=h(R_j\oplus R_i^*\oplus R_c\oplus(f_i\ ||f_j))$ and replaces $(A_1, CSID_j)$ with $(A_1^{new}, CSID_j^{new})$. Next, it generates timestamp T_s^* and sends the message $M_4=\{\alpha, Z_2, N_8, N_9, N_{10}\}$ to user as shown in Fig. 3.

***Step*-5**: The U_i computes $f_i=h(R_i||ID_u||HW_u)$, $f_j^*=\alpha\oplus f_i$, $R_j^*\oplus R_c=N_8\oplus h(R_i||HW_u)$, $Sk_u=h(R_i\oplus R_j^*\oplus R_c\oplus(f_i||f_j))$, $CID_i^{new}=h((R_j^*\oplus R_c)||\ R_i)\ \oplus N_9$, $N_1^{new}=N_{10}\oplus h(R_i\ ||(R_j^*\oplus R_c))$, $Z_2=h(N_1^{new}||CID_i^{new}||N_4||ID_u||f_i\ ||f_j\ ||\ R_i)$,. Then it verifies the equality for $Z_2^*\ ?{=}\ Z_2$. If it is not true, it abandons the session. Otherwise, calculates $E_1^{new}=N_1^{new}\oplus h(ID_u\ ||\ HW_u)$ and replaces (E_1, CID_i) with (E_1^{new}, CID_i^{new}) in its smart card.

IV. SECURITY ANALYSIS

This section presents the formal analysis, ProVerif-oriented security verification, and informal discussion on security aspects.

A. FORMAL ANALYSIS (BAN LOGIC)

We use BAN logic to analyze that the participants Ui and Sj mutually share the created session key SK as computed by the CA. Using this SK, the user can get the desired data from the server. The following symbols and procedures for this analysis can be referred to [29], [32]-[33].

a) Goals

We lay down the following Target Goals (TG):

TG_1 : $U|\equiv U\overset{SK}{\leftrightarrow}S$

TG_2 : $S|\equiv U\overset{SK}{\leftrightarrow}S$

TG_3 : $CA|\equiv U\overset{SK}{\leftrightarrow}S$

TG_4 : $U|\equiv S|\equiv U\overset{SK}{\leftrightarrow}S$

TG_5 : $S|\equiv U|\equiv U\overset{SK}{\leftrightarrow}S$

TG_6 : $CA|\equiv U|\equiv U\overset{SK}{\leftrightarrow}S$

TG_7 : $CA|\equiv S|\equiv U\overset{SK}{\leftrightarrow}S$

b) Message Idealization

M_1: $U\rightarrow S$: $\{CID_i, N_2, N_3, N_4, T_u\}$

M_2: $U\rightarrow CA$: $\{CID_i, N_2, N_3, N_4\}$

M_3: $S\rightarrow CA$: $\{CID_i, N_2, N_3, N_4, CSID_j, N_5, N_6, N_7, T_s\}$

M_4: $CA\rightarrow U$: $\{\alpha, Z_2, N_8, N_9, N_{10}, T_s^*, T_c\}$

M_5: $CA\rightarrow S$: $\{\alpha, Z_1, Z_2, N_8, N_9, N_{10}, N_{11}, N_{12}, N_{13}, T_c\}$

M_6: $S\rightarrow U$: $\{\alpha, Z_2, N_8, N_9, N_{10}, T_s^*\}$

c) Preliminary Assumptions of States

A1 : $U\ |\equiv\sharp(R_i)$

A2 : $S\ |\equiv\sharp(R_j)$

A3 : $CA\ |\equiv\sharp(R_c)$

A4 : $CA\ |\equiv U\overset{h(CIDi\,||\,y)}{\longleftrightarrow}CA$

A5 : $CA\ |\equiv\sharp(CID_i)$

A6 : $CA|\equiv\sharp(CSID_i)$

A7 : $CA|\equiv U|\Rightarrow R_i$

A8 : $CA|\equiv S|\Rightarrow N_j$

A9 : $CA|\equiv U|\Rightarrow ID_u$

A10 : $CA|\equiv S|\Rightarrow ID_j$

A11 : $CA|\equiv\sharp(ID_u)$

A12 : $CA|\equiv\sharp(SID_j)$

A13 : $U\ |\equiv U\overset{HW_u}{\longleftrightarrow}CA$

A14 : $CA\ |\equiv U\overset{HW_u}{\longleftrightarrow}CA$

A15 : $CA\ |\equiv S\overset{h(CSIDj\,||\,y)}{\longleftrightarrow}CA$

A16 : $CA\ |\equiv S\overset{e_s}{\leftrightarrow}CA$

A17 : $S\ |\equiv S\overset{e_s}{\leftrightarrow}CA$

A18 : $U\ |\equiv U\overset{f_i}{\leftrightarrow}CA$

A19 : $U|\equiv CA|\Rightarrow f_j$

A20 : $S|\equiv CA|\Rightarrow f_i$

A21 : $U|\equiv\sharp(R_j\oplus R_c)$

A22 : $U|\equiv CA|\Rightarrow(R_j\oplus R_c)$

A23 : $S|\equiv S\overset{f_j}{\leftrightarrow}CA$

A24: $S|\equiv\sharp(R_i\oplus R_c)$

A25 : $S|\equiv CA|\Rightarrow(R_i\oplus R_c)$

A26 : $S|\equiv S\overset{h(CSID_j\,||\,y)}{\longleftrightarrow}CA$

A27 : $U|\equiv U\overset{h(CID_j\,||\,y)}{\longleftrightarrow}CA$

A28: $S|\equiv\sharp(CID_j)$

A29 : $S|\equiv U|\Rightarrow R_i$

A30: $CA|\equiv\sharp(R_i)$

A31: $CA|\equiv\sharp(R_j)$

d) Proof

Referring to M_1, and Seeing Rule (S-R)

S1: $S\triangleleft\{CID_i, N_2{:}\langle R_i, CID_i\rangle_y, N_3, N_4, T_u\}$

Using S1, we have

S2: $S\triangleleft\{\langle R_i, CID_i\rangle_y\}$

Employing A26, A27, we have

S3: $S|\equiv S\overset{h(CID_j\,||\,y)}{\longleftrightarrow}U$

Employing S2, S3, as well as Message-Meaning (M-M) rule, we have

S4: $S\ |\equiv U\ |\sim(R_i, CID_i)$

Referring to A28, S4, Freshness Rule (F-R), and the nonce verification rule, we have

S5: $S\ |\equiv U\ |\equiv(R_i, CID_i)$

After applying on each statement, we have

S6: $S\ |\equiv U\ |\equiv R_i$

Applying A29, S6, and Jurisdiction Rule (J-R), we have

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381616

S7: S |≡ R_i

Now considering M_2, applying the S-R, we have

```
(****************** Sj's process******************)
Let process_Sj=
new SIDj: bitstring;
new es: bitstring;
out(S_Cl, (es, SIDj));
in(S_Cl,(zCSIDj:bitstring, zA1:bitstring);
!
in(P_Cl,(zCIDi:bitstring, zN2:bitstring, zN3:bitstring, zN4:bitstring, zTu:bitstring));
new Rj:bitstring;
new A1: bitstring;
Let N5=XOR(A1, Rj) in
Let N6=XOR(h(SIDj, Rj), SIDj) in
New Ts:bitstring;
Let N7=h(con(con(Ri, SIDj)), con(zN4, Ts) in
out(P_Cl,(zCIDi, zCSIDj, zN2, zN3, zN4, N5, N6, N7, zTu, Ts));
in(P_Cl,(zZ1: bitstring, zZ3: bitstring, zN11: bitstring, zN12: bitstring, zN13: bitstring, zTc:bitstring)
let zfj=h(con, con(SIDj, Rj), es) in
let zfi'= XOR (, zfj) in
let CSIDjnew=XOR(h(con(XOR(zN11, h(con(Rj, es))), Rj)), zN12) in
let A1new=XOR(zN13, h(con(Rj, XOR(zN11, h(con(Rj, es)))))) in
let zZ1'=h(con(con(con(A1new, CSIDjnew), con(zfj, N7)), con(con(Rj, SIDj), con(zfi', Tc)))) in
if zZ1'= zZ1 then
let Q'=h(con(con(con(zfj, N7), con(Rj, SIDj)), con(zfi, Tc)) in
let substance=sdec(zZ3, Q') in
let Tc=getsecond(substance) in
let N10=getsecond(getfirst(substance)) in
let N9= getsecond(getfirst(getfirst(substance))) in
let N8= getsecond(getfirst(getfirst(getfirst(substance)))) in
let Z2= getfirst(getfirst(getfirst(substance))) in
let SKs= h(XOR(XOR(Rj, XOR(zN11, h(con(Rj, es)))), con(zfi, zfj))) in
let CSIDj=CSIDjnew in
let A1=A1new in
out(P_Cl,(, Z2, N8, N9, N10);
event Ui_Authed();
0
).
```

Fig. 6 Server's process

S8: CA ⊲{CID_i, N_2: $\langle R_i, CID_i\rangle_y$, N_3: $\langle ID_u\rangle_{h(CID_i||R_i)}$, N_4, T_u}

According to S-R,

S9: CA ⊲{$\langle R_i, CID_i\rangle_y$ }

Applying A4, S9, and M-M rule, we have

S10: CA|≡U|~ (R_i, CID_i)

Next, according to A5, S3, the F-R & N-V rule, we have

S11: CA|≡U|≡(R_i, CID_i)

By applying S11 property and Belief Rule (B-R), we have

S12: CA |≡U|≡(R_i)

S13: CA |≡U|≡(CID_i)

By applying A7, S12, and J-R, we have

S14: CA| ≡R_i

Using S8 and the S-R, we have

S15: CA ⊲ {$\langle ID_u\rangle_{h(CID_i||R_i)}$ }

Applying A5, S14, & M-M rule, we have

S16:CA|≡U|~ (ID_u)

Applying A11, S16, & N-V rule, we have

S17: CA|≡U|≡ID_u

Applying A9, S17, and J-R, we have

S18: CA|≡ ID_u

Applying A14, S14, S18, and the B-R, we have

S19: CA |≡ (ID_u, R_i, HW_u)

∵ $f_i = h(R_i \,||\, ID_u \,||\, HW_u)$, we have

S20: CA |≡ f_i

Using M_3 as well as the S-R, we have

S21: CA⊲{$CSID_j$, N_5: $\langle R_j, CSID_j\rangle_y$, N_6: $\langle SID_j\rangle_{h(CSID_j||R_j)}$, N_7, T_s}

On the application of S-R, we have

S22: CA ⊲{$\langle R_j, CSID_j\rangle_y$ }

Applying A15, S22, and M-M rule, we have

S23: CA|≡ S|~ $(R_j, CSID_j)$

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381616

Applying A6, S23, the F-R, and N-V rule, we have

S24: CA |≡ S | ≡$(R_j, CSID_j)$

Applying the B-R, we have

S25: CA |≡ S | ≡ R_j

S26: CA |≡ S | ≡ $CSID_j$

Applying A8, S25, and J-R, we have

S27: CA | ≡R_j

Using S21, and S-R, we have

S28: CA ⊲ $\{\langle SID_j\rangle_{h(CSID_j || R_j)}\}$

Applying S27, CA ⊲$CSID_j$, and M-M rule, we have

S29: CA |≡S | ~SID_j

Applying A12, S29, and N-V rule, we have

S30: CA |≡ S |≡ SID_j

Applying A10, S30, and J-R, we have

S31: CA |≡ SID_j

Applying A16, S31, S27, and B-R, we have

S32: CA |≡ (SID_j, R_j, e_s)

∵$f_j = h(SID_j || R_j || e_s)$, we have

S33: CA |≡f_j

Applying A3, S14, S20, S27, S33, and B-R, we have

S34: CA |≡ U$\overset{SK}{\leftrightarrow}$S (**TG₃**)

Applying A30, S34, and Session Key (S-K) rule, we have

S35: CA|≡U |≡U$\overset{SK}{\leftrightarrow}$S (**TG₆**)

Referring A31, S34, and S-K rule, we have

S36: CA |≡S |≡U$\overset{SK}{\leftrightarrow}$S (**TG₇**)

Using M4, and S-R, we have

S37: U⊲{ α: , N_5: $\langle f_j\rangle_{f_i}$; Z_2: $\langle N_4, ID_u, f_j, R_i, T_c\rangle_{f_i}$; N_8: $\langle R_j \oplus R_c\rangle_{h(R_i || HW_u)}$; T_c}

Applying the S-R, we have

S38: U⊲{$\langle N_4, ID_u, f_j, R_i, T_c\rangle_{f_i}$}

Applying A18, S38, and M-M rule, we have

S39: U|≡CA | ~ (N_4, ID_u, f_j, R_i, T_c)

Applying A1, S39, F-R and N-V rule, we have

S40: U |≡ CA |≡ (N_4, ID_u, f_j, R_i, T_c)

Utilizing the B-R, we have

S41: U |≡ CA |≡f_j

Applying A19, S41, and N-V rule, we have

S42:U |≡f_j

Referring to S37 and S-R, we have

S43: U⊲{$\langle R_j \oplus R_c\rangle_{h(R_i || HW_u)}$}

Applying A1, A13, S14, as well as M-M rule, we have

S44: U|≡ CA | ~ ($R_j \oplus R_c$)

Applying A21, S44, and N-V rule, we have

S45: U|≡ CA | ≡ ($R_j \oplus R_c$)

Applying A22, S45, and J-R, we have

S46: U|≡ ($R_j \oplus R_c$)

Applying A1, S48, and S-K rule, we have

S49: U |≡S |≡ U$\overset{SK}{\leftrightarrow}$S (**TG₄**)

Referring M_5, and S-R, we have

S50: S ⊲{ α: $\langle f_i\rangle_{f_j}$; Z_1: $\langle N_7, SID_j, f_i, R_j, T_c\rangle_{f_j}$; N_9: $\langle R_j \oplus R_c\rangle_{h(R_j || e_s)}$; T_c}

Applying the S-R, we have

S51: S ⊲{$\langle N_7, SID_j, f_i, R_j, T_c\rangle_{f_j}$}

Applying A23, S51, and M-M rule, we have

S52: S|≡ CA | ~ ($N_7, SID_j, f_i, R_j, T_c$)

Applying A2, S52, N-V rule and freshness rule, we have

S53: S |≡ CA |≡ ($N_7, SID_j, f_i, R_j, T_c$)

Utilizing the F-R, we have

S54: S |≡ CA |≡ f_i

Applying A20, S54, and N-V rule, we have

S55:S |≡ f_i

In accordance with S50 and S-R, we have

S56: S⊲{$\langle R_i \oplus R_c\rangle_{h(R_j || e_s)}$}

Applying A2, A17, S33, and M-M rule, we have

S57:S|≡ CA | ~ ($R_i \oplus R_c$)

Applying A24, S57, and N-V rule, we have

S58: S|≡ CA |≡ ($R_i \oplus R_c$)

Applying A25, S58, and J-R, we have

S59: S|≡ ($R_i \oplus R_c$)

Applying A2, A23, S55, S58, and F-R, we have

S60: S|≡ ($R_j, R_i \oplus R_c, f_i, f_j$)

S61: S |≡ U$\overset{SK}{\leftrightarrow}$S (**TG₂**)

Applying A2, S61, and S-K rule, we have

S62: S|≡U| ≡ U $\overset{SK}{\leftrightarrow}$S (**TG₅**)

B. PROVERIF TOOL-BASED VERIFICATION

In this section, we used automated ProVerif [38] to validate the security properties of the proposed model. It helps to prove authentication, secrecy as well as observational equivalence features of the cryptographic techniques. It provides support of frequently employed crypto-operations to construct the protocols. It translates the protocol algorithm into abstract illustration using Horn clauses, and evaluates the probability of security fetures held by resolution on the clauses. We designed a simulation model based on the processes for the user, cloud server, control authority registration, as well as mutual authentication. The following procedures are demonstrated in ProVerif:

1) We created secure and public channels, i.e., *S_Cl* and *P_Cl,* for the purpose of communication in registration and mutual authentication phases, respectively. Using secure channel, the user and cloud servers are registered, while the public channel is used to establish session keys SK_u, SK_s and SK_c by the user, cloud server, and control authority, respectively. We also employed XOR, hash function, and string connection operations in different procedures.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3381616


```
(***************** CA’s process*****************)
Let Ui_reg=
in(S_Cl,(rIDu:bitstring, rHWu:bitstring));
new CIDi:bitstring;
let Ci=h(XOR(h(rIDu), rHWu)) in
let E1=XOR(h(con(CIDi, y)), h(con(rIDu, rHWu))) in
out(S_Cl, (CIDi, Ci, E1, no));

let Sj_Reg=
in(S_Cl,(res:bitstring, rSIDj:bitstring));
new CSIDj:bitstring;
let A1=h(con(CSIDj, y)) in
out(S_Cl, (CSIDj, A1));

let CA_Auth=
in(P_Cl,(yCIDi:bitstring, yCSIDj:bitstring, yN2:bitstring, yN3:bitstring, yN4:bitstring, yN5:bitstring,
yN6:bitstring, yN7:bitstring, yTu:bitstring, yTs:bitstring));
let Ri’=XOR(yN2, h(con(yCIDi, y))) in
let IDu’=XOR(yN3, h(con(yCIDi, Ri’))) in
let N4’=h(con(con(con(Ri’, yCIDi ), IDu’), yTu)) in
if N4’= yN4 then
let Rj’=XOR(yN5, h(con(SIDj, y))) in
let SIDj’=XOR(yN6, h(con(yCSIDj, Rj’))) in
let N7’=h(con(con(con(Rj’, SIDj’ ), N4’), yTs)) in
if N7’= yN7 then
new Rc:bitstring;
new CIDnew:bitstring;
new CSIDnew:bitstring;
new Tc:bitstring;
let fi=h(con(Ri’, IDu’), rHWu) in
let fj=h(con(SIDj’, Rj’), res) in
let y =XOR(fi, fj) in
let SKc=h(XOR(XOR(XOR(Ri’, Rj’), Rc), con(fi, fj)) in
let N8=XOR(XOR(Rj’, Rc), h(con(Ri’, rHWu))) in
let N9=XOR(h(con(XOR(Rj’, Rc), Ri’)), CIDinew) in
let N10=XOR(h(CIDinew, y), h(con(Ri’, XOR(Rj’, Rc)))) in
let N11=h(XOR(XOR(Ri’, Rc), h(con(Rj’, res)))) in
let N12=XOR(h(con(XOR(Ri’, Rc), Rj’)), CSIDjnew) in
let N13=XOR(h(con(CSIDjnew, y), h(con(Rj’, XOR(Ri’, Rc))))) in
let Z1=h(con(con(con(con(con(con(con(h(con(CSIDjnew, y)), CSIDjnew), fi), yN7), Rj’), SIDj), fi), Tc)) in
let Z2=h(con (con(con(con(con(con(h(con(CIDinew, y)), CIDinew), yN4), rIDu’), fi), fj), Ri’)) in
let Q= h(con (con(con(con(con(fj, yN7), Rj’), SIDj), fi), Tc)) in
let substance= con(con(con(con(Z2, N8), N9), N10), Tc) in
let Z3=senc(substance, Q) in
out(P_Cl, (, Z1, Z3, N11, N12, N13, Tc));
0.
Let process_CA = Ui_reg | Sj_reg | CA_Auth.
```

Fig.7 CA’s Process

2) We initiated a few queries for validating the security requirements. Fig. 4 describes the channels, constraints, variables, and constructors. Fig. 5 depicts the user’s process as process_Ui. Fig. 6 describes the cloud server’s process as process_Sj, while Fig. 7 shows the process of control authority CA. In Fig. 7, we describe the protocol by employing User_Authed(.) and User_Started(.).

3) The tested results are

RESULT not attacker(SKu[]) is true.
RESULT not attacker(SKs[]) is true.
RESULT not attacker(SKc[]) is true.
RESULT inj-event(Ui_authed) ==>inj-event(Ui_started) is true.

Hence, the contributed model affirms the verification with ProVerif by defeating the known attacks with the help of mutually established session keys, i.e., SK_u, SK_s, and SK_c.

C. INFORMAL SECURITY ANALYSIS

This subsection discusses the security aspects of the proposed scheme on informal lines, as given below:

1) MUTUAL AUTHENTICATION

In SEMS-5G, all of the three participants mutually authenticate one another in a single session [39]. If an adversary eavesdrop the communication messages $\{M_1\text{-}M_4\}$ on a public channel, it will not be able to calculate the mutually agreed session key *SK* as constructed between the legal entities. This is because, the server S_j verifies user U_i and *CA* on the basis of verification for equality $Z_1 ?= h(A_1^{new} || CSID_j^{new} || f_j || N_7 || R_j || SID_j || f_i^* || T_c)$. If it is not true, there must be some possibility of replay or impersonation attack on the part of the adversary, and S_j shall abort the session. S_j monitors the shared parameter e_s to verify a message from *CA*, however, it cannot authenticate U_i directly, though it can verify that *CA* responds to the same authentication request that is forwarded to *CA*. The *CA* authenticates U_i and S_j on the positive verification reports of the equations N_4^* ?=N_4 and N_7^* ?=N_7, respectively, by consulting verifiers in its repository. These verifications justify the issuance of pseudonym identities CID_i and $CSID_j$ by the CA. Likewise, U_i verifies S_j as well as CA upon the positive verification of equality Z_2 ?=$h(N_4 \| ID_u^*)\|f_i \|f_j \| R_i^*)$. This equation matching entails that Z_2 is constructed by legal CA while f_i can only produce by that CA having HW_u, and the message is forwarded by the same S_j whom it had forwarded the authentication request.

2) RESISTS OFFLINE-PASSWORD GUESSING THREAT

In SEMS-5G, the attacker may not compute the password or identity since the user computes D_u by calculating $D_u=h(ID_u \| PW_u)$ $mod\ n_0$, while the D_u parameter is stored as $E_2= r_u \oplus D_u$ in the smart card. This eliminates the chances of guessing either low entropy password or identity [34] since there will be $\frac{|\mathcal{F}_{ID_u} * \mathcal{F}_{PW_u}|}{n} \approx 2^{32}$ candidate (ID_u, PW_u) pairs satisfying C_i^* ?=C_i, where $\mathcal{F}$ represents the possible combinations in identity ID_u or password PW_u.

3) RESISTS REPLAY OR IMPERSONATION ATTACK

If an attacker intercepts the messages $\{M_1\text{-}M_4\}$ on a public channel, and replay any of these messages towards legal participants Ui, Sj, or CA, then these entities may successfully foil the replay attack with conviction [40]. The CA may assess the possibility of replay by verifying the timestamp as well as the equation N_7^* ?=N_7. The server and user may thwart this attack by verifying the equations $Z_1^*=Z_1$ and $Z_2^*=Z_2$, respectively. Although the server and user do not verify CA on the basis of timestamp, yet these entities may counter the replay attack on the basis of nonce verification.

4) SUPPORTS PERFECT FORWARD SECRECY

In SEMS-5G, if an attacker manages to compromise the private secret key of a legal participant, yet the former may not be able to compute a valid session key $SK_u= SK_s=SK_c= h(R_i \oplus R_j \oplus R_c \oplus (f_i \| f_j))$. This is because of the fact; the users and servers share other crucial parameters as well, for instance HW_u and e_s, respectively, with CA contributing towards upholding the feature of perfect forward secrecy.

5) SUPPORTS ANONYMITY

The SEMS-5G confers anonymity and untraceability-based security features to the user. An adversary may neither recover identity ID_u of the user, nor could have any clue using the intercepted messages $\{M_1\text{-}M_4\}$ that may help the former to associate a session to a particular user U_i [41]. An adversary cannot break the assumption of hash digest function and guess the original identity ID_u from the recovered C_i or E_1 parameters in polynomial amount of time.

6) RESISTS MAN-IN-THE-MIDDLE ATTACK

The SEMS-5G, unlike Wu *et al.*, ensures mutual authentication to the involved participants as elaborated in sub-section (1) above, which nullifies any probability for an adversary to initiate MIDM attack [42].

7) RESISTS PRIVILEGED INSIDER ATTACK

Our scheme is resistant to privileged insider attacks in case the attacker is able to compromise the registration request parameters of the user during the registration phase [43]. For instance, the adversary after accessing the registration request parameters ID_u, HW_u may not compromise the session key. For guessing the password from HW_u, the adversary needs to break the assumption of hash digest function. It is computationally infeasible to compute the password in polynomial amount of time without the knowledge of random number r_u. For this purpose, the adversary must compromise the smart card and steal its contents as well [47]. Therefore, our scheme may resist a privileged insider attack.

8) RESISTS STOLEN VERIFIERS ATTACK

In SEMS-5G, if $\mathcal{A}$ steals the users' verifiers from the repository of CA, for instance HW_u, the former will not be able to calculate mutually authenticated session key as constructed among the legal participants [49]. In our scheme, the calculation of session key SK must require access to Ui's smart card, shared secrets, and access to the private secret key of CA at the same time, which is based on the strong assumption of the capabilities related to the adversary [50].

9) RESISTS STOLEN SMART CARD ATTACK

Table III: Functionality Comparison

	Irshad *et al.* [29]	Amin *et al.* [31]	Wu *et al.* [30]	Wu *et al.*[35]	SEMS-5G
Achieves anonymity and untraceability	×	×	✓	✓	✓
Immune to an offline-password guessing attack	✓	×	✓	×	✓
Immune to an impersonation attack	×	✓	×	×	✓
Immune to replay attack	×	✓	✓	✓	✓
Resists session-specific temporary information attack	✓	✓	×	✓	✓
Immune to stolen smart card attack	✓	✓	×	✓	✓
Immune to Privileged insider attack	×	✓	×	✓	✓
Achieves session key agreement	✓	×	✓	×	✓
Achieves mutual authentication	×	✓	✓	×	✓
Achieves perfect forward secrecy	✓	✓	×	✓	✓

Table IV. Computational cost (ms)

	Irshad *et al.* [29]	Amin *et al.* [31]	Wu *et al.* [30]	Wu *et al.* [35]	SEMS-5G
U_i	$4T_H$+$3T_C$≈ *381.14ms*	$9T_H$≈ *0.046ms*	$11T_H$≈ *0.056ms*	$13T_H$≈ *0.068ms*	$14T_H$≈ *0.072ms*
S_j	$4T_H$+$2T_C$≈ *254.10ms*	$4T_H$≈ *0.020ms*	$6T_H$≈ *0.031ms*	$8T_H$≈ *0.041ms*	$9T_H$ +T_S≈ *0.056ms*
CA	$6T_H$+T_C≈ *127.07ms*	$10T_H$≈ *0.051ms*	$19T_H$≈ *0.098ms*	$19T_H$≈ *0.098ms*	$20T_H$+T_S≈ *0.113ms*
Total	$14T_H$+$6T_C$≈ *762.32ms*	$23T_H$≈ *0.119ms*	$36T_H$≈ *0.186ms*	$40T_H$≈ *0.206ms*	$43T_H$+$2T_S$≈ *0.243ms*

If an attacker steals the user's smart card and its contents $\{C_i, E_1, E_2, CID_i, n_0, h(.)\}$, still the former may not be able to initiate either impersonation attack or compute previous session keys. This is because; the attacker has no access to either HW_u or identity ID_u or password PW_u parameters. Hence, our scheme is immune to stolen smart card attacks.

V. PERFORMANCE ANALYSIS

The formal and informal analysis demonstrates the security strength of the contributed scheme over previously presented schemes. In this section, we evaluate and analyze the performance of proposed scheme against various multiserver authentication schemes such as Irshad *et al*. [29], Amin *et al*. [31], Wu *et al*. [30], Wu *et al*. [35] in distributed cloud-based 5G environment. Table III depicts the comparisons of security functions for various authentication schemes with our proposed scheme. As it is evident from that table, the schemes [29] and [31] does not support anonymity and untraceability features. The schemes [31] and [35] are prone to offline-password guessing attacks. Similarly, [29], [30] and [35] are susceptible to impersonation attacks. The Wu *et al*. scheme [30] is found to be prone to stolen smart card attacks and session-specific temporary information attacks. The schemes [29] and [30] do not provide resistance to privileged insider attacks. Similarly, the schemes [29] and [35] lack mutual authentication between legal participants, while the [30] does not support perfect forward secrecy to its stakeholders.

Table IV depicts the comparison of computational costs of our scheme and other related protocols. We assumed the computational costs of the scheme [30]. According to [30], the computational delay of the user, server, and CA in our scheme can be computed as 0.072ms, 0.056ms, and 0.113ms, respectively. The total computational delay is calculated as 0.243ms. The computational cost of our scheme is less than Irshad *et al*. [29], while a little more than [30], [31], and [35], i.e., 0.119ms, 0.186ms, and 0.206ms respectively. Although, the cost of our scheme is a bit more than [30]-[31, [35] with few more hash-based operations, yet more secure than those schemes in terms of security features as depicted from Table III. In our scheme, the exclusive-OR operation is assumed to be of negligible cost [14]. Hence, the security assessment regarding Table III shows that the contributed scheme is not only immune to all known attacks, unlike previous schemes, but bears almost the same computational cost, as a few more hash operations bear trivial additional cost, but enhances the security of scheme as depicted in tables.

VI. CONCLUSION

The paradigm shift from a conventional centralized system towards 5G-based distributed network domains raises many security challenges. In this paper, we critically examine Wu *et al*., a multiserver authentication scheme that was proposed for a distributed cloud-oriented 5G environment. The scheme supports few convincing security properties including perfect forward secrecy and anonymity. However, the scheme is vulnerable to a man-in-the-middle attack, impersonation attack, as well as offline password guessing attack. Hence the identified threats render the scheme impractical for industrial applications. In this context, we proposed an efficient and secure multiserver authentication protocol (SEMS-5G) for distributed cloud-based 5G architecture, ensuring PFS, anonymity, untraceability, as well as resistance to all known attacks. The formal security analysis and automated simulation-based validated results prove that our scheme is efficient as well as address the concerns that earlier protocols could not address in 5G cloud based architecture.

Acknowledgment

Researchers would like to thank the Deanship of Scientific Research, Qassim University for funding publication of this project.

REFERENCES

[1] Nencioni, G., Garroppo, R. G., & Olimid, R. F. (2023). 5G Multi-access Edge Computing: a Survey on Security, Dependability, and Performance. IEEE Access.
[2] Salahdine, F., Han, T., & Zhang, N. (2023). Security in 5G and beyond recent advances and future challenges. Security and Privacy, 6(1), e271.
[3] Lai, C., Ma, Y., Lu, R., Zhang, Y., & Zheng, D. (2022). A novel authentication scheme supporting multiple user access for 5G and beyond. IEEE Transactions on Dependable and Secure Computing.
[4] Miao, J., Wang, Z., Ning, X., Xiao, N., Cai, W., & Liu, R. (2022). Practical and secure multifactor authentication protocol for autonomous vehicles in 5G. Software: Practice and Experience.
[5] Zhang, J., Zhong, H., Cui, J., Tian, M., Xu, Y., & Liu, L. (2020). Edge Computing-based Privacy Preserving Authentication Framework and Protocol for 5G-enabled Vehicular Networks. IEEE Transactions on Vehicular Technology
[6] Hsu, C. L., Le, T. V., Lu, C. F., Lin, T. W., & Chuang, T. H. (2020). A Privacy-Preserved E2E Authenticated Key Exchange

Protocol for Multi-Server Architecture in Edge Computing Networks. IEEE Access, 8, 40791-40808

[7] B. Ying and A. Nayak, ‘‘Lightweight remote user authentication protocolfor multiserver 5G networks using self-certified public key cryptography,’’*J. Netw. Comput. Appl.*, vol. 131, pp. 66–74, Apr. 2019.

[8] E. Borcoci, T. Ambarus, J. Bruneau-Queyreix, D. Negru, and J. M. Batalla,‘‘Optimization of multiserver video content streaming in 5G environment,’’ in *Proc. 8th Int. Conf. Evolving Internet*, Barcelona, Spain, 2016.

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

[10] T.-Y. Hwang, ‘‘Password authentication using public-key encryption,’’ in*Proc. Int. Carnahan Conf. Secur. Technol.*, 1983, pp. 35–38.

[11] S.-P. Shieh, W.-H. Yang, and H.-M. Sun, ‘‘An authentication protocol without trusted third party,’’ *IEEE Commun. Lett.*, vol. 1, no. 3, pp. 87–89, May 1997.

[12] M. Sandirigama, A. Shimizu, and M. T. Noda, ‘‘Simple and secure password authentication protocol (SAS),’’ *IEICE Trans. Commun.*, vol. E83-B,no. 6, pp. 1363–1365, 2000.

[13] C.-C. Chang and T.-C. Wu, ‘‘Remote password authentication with smartcards,’’ *IEICE Proc. E, Comput. Digit. Techn.*, vol. 138, no. 3, pp. 165–168,1991.

[14] M.-S. Hwang and L.-H. Li, ‘‘A new remote user authentication scheme using smart cards,’’ *IEEE Trans. Consum. Electron.*, vol. 46, no. 1,pp. 28–30, Feb. 2000.

[15] C.-C. Chang and K.-F. Hwang, ‘‘Some forgery attacks on a remote user authentication scheme using smart cards,’’ *Informatica*, vol. 14, no. 3,pp. 289–294, 2003.

[16] S.-W. Lee, H.-S. Kim, and K.-Y. Yoo, ‘‘Improvement of Chien's remote user authentication scheme using smart cards,’’ *Comput. Standards Interfaces*, vol. 27, no. 2, pp. 181–183, 2005.

[17] B.-L. Chen, W.-C. Kuo, and L.-C. Wuu, ‘‘Robust smart-card-based remote user password authentication scheme,’’ *Int. J. Commun. Syst.*, vol. 27, no. 2,pp. 377–389, Feb. 2014.

[18] X. Li, J. Niu, M. Khurram Khan, and J. Liao, ‘‘An enhanced smart cardbased remote user password authentication scheme,’’ *J. Netw. Comput.Appl.*, vol. 36, no. 5, pp. 1365–1371, Sep. 2013.

[19] L.-H. Li, L.-C. Lin, and M.-S. Hwang, ‘‘A remote password authentication scheme for multiserver architecture using neural networks,’’ *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1498–1504, Nov. 2001.

[20] S. Ghosh, A. Majumder, J. Goswami, A. Kumar, S. P. Mohanty, andB. K. Bhattacharyya, ‘‘Swing-pay: One card meets all user payment and identity needs: A digital card module using NFC and biometric authentication for peer-to-peer payment,’’ *IEEE Consum. Electron. Mag.*, vol. 6,no. 1, pp. 82–93, Jan. 2017.

[21] C.-C. Lee, T.-H. Lin, and R.-X. Chang, ‘‘A secure dynamic ID based remote user authentication scheme for multiserver environment using smart cards,’’ *Expert Syst. Appl.*, vol. 38, no. 11, pp. 13863–13870, 2011.

[22] X. Li, Y. Xiong, J. Ma, and W. Wang, ‘‘An efficient and security dynamic identity based authentication protocol for multiserver architecture using smart cards,’’ *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 763–769,Mar. 2012.

[23] K. Xue, P. Hong, and C. Ma, ‘‘A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multiserver architecture,’’ *J. Comput. Syst. Sci.*, vol. 80, no. 1,pp. 195–206, Feb. 2014.

[24] S. Jangirala, S. Mukhopadhyay, and A. K. Das, ‘‘A multiserver environment with secure and efficient remote user authentication scheme basedon dynamic ID using smart cards,’’ *Wireless Pers. Commun.*, vol. 95, no. 3,pp. 2735–2767, Aug. 2017.

[25] S. Shunmuganathan, R. D. Saravanan, and Y. Palanichamy, ‘‘Secure and efficient smart-card-based remote user authentication scheme for multiserver environment,’’ *Can. J. Elect. Comput. Eng.*, vol. 38, no. 1, pp. 20–30,2015.

[26] H. Zhu, ‘‘Flexible and password-authenticated key agreement schemebased on chaotic maps for multiple servers to server architecture,’’ *WirelessPers. Commun.*, vol. 82, no. 3, pp. 1697–1718, Jun. 2015.

[27] J.-L. Tsai and N.-W. Lo, ‘‘A privacy-aware authentication scheme fordistributed mobile cloud computing services,’’ *IEEE Syst. J.*, vol. 9, no. 3,pp. 805–815, Sep. 2015.

[28] H. Xiong and Z. Qin, ‘‘Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks,’’*IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1442–1455, Jul. 2015.

[29] A. Irshad, H. F. Ahmad, B. A. Alzahrani, M. Sher, and S. A. Chaudhry,‘‘An efficient and anonymous chaotic map based authenticated key agreement for multiserver architecture,’’ *KSII Trans. Internet Inf. Syst.*, vol. 10,no. 12, pp. 5572–5595, 2016.

[30] F. Wu, X. Li, L. Xu, A. K. Sangaiah, and J. J. Rodrigues, ‘‘Authentication protocol for distributed cloud computing: An explanation of the security situations for Internet-of-Things-enabled devices,’’ *IEEE Consum. Electron. Mag.*, vol. 7, no. 6, pp. 38–44, Nov. 2018.

[31] R. Amin, N. Kumar, G. Biswas, R. Iqbal, and V. Chang, ‘‘A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment,’’ *Future Gener. Comput. Syst.*, vol. 78, pp. 1005–1019,Jan. 2018.

[32] M. Burrows, M. Abadi, and R. M. Needham, ‘‘A logic of authentication, ’’*Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 426, no. 1871, pp. 233–271,1989

[33] R. Amin and G. Biswas, ‘‘A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks,’’*Ad Hoc Netw.*, vol. 36, pp. 58–80, Jan. 2016.

[34] Wang, D., Wang, P., & Wang, C. (2020). Efficient multi-factor user authentication protocol with forward secrecy for real-time data access in WSNs. *ACM Transactions on Cyber-Physical Systems*, *4*(3), 1-26

[35] Wu, T. Y., Lee, Z., Obaidat, M. S., Kumari, S., Kumar, S., & Chen, C. M. (2020). An authenticated key exchange protocol for multiserver architecture in 5G networks. *IEEE Access*, *8*, 28096-28108

[36] Kunal, S., Saha, A., & Amin, R. (2019). An overview of cloud-fog computing: Architectures, applications with security challenges. Security and Privacy, 2(4), e72.

[37] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, "Security and privacy challenges in mobile cloud computing: Survey and way ahead," *J. Netw. Comput. Appl.*, vol. 84, pp. 38–54, 2017

[38] Blanchet B. ProVerif Automatic Cryptographic Protocol Verifier User Manual. Paris, France: Departement d Informatique, Ecole Normale Superieure, CNRS; 2005.

[39] Karim, S. M., Habbal, A., Chaudhry, S. A., & Irshad, A. (2023). BSDCE-IoV: Blockchain-Based Secure Data Collection and Exchange Scheme for IoV in 5G Environment. IEEE Access.

[40] Lai, C., Ma, Y., Lu, R., Zhang, Y., & Zheng, D. (2022). A novel authentication scheme supporting multiple user access for 5G and beyond. *IEEE Transactions on Dependable and Secure Computing*.

[41] Cho, Y., Oh, J., Kwon, D., Son, S., Yu, S., Park, Y., & Park, Y. (2022). A secure three-factor authentication protocol for e-

governance system based on multiserver environments. *IEEE Access*, 10, 74351-74365.
[42] Sahoo, S. S., Mohanty, S., Sahoo, K. S., Daneshmand, M., & Gandomi, A. H. (2023). A Three Factor based Authentication Scheme of 5G Wireless Sensor Networks for IoT System. *IEEE Internet of Things Journal*
[43] Wang, J., Wu, L., Wang, H., Choo, K. K. R., Wang, L., & He, D. (2022). A Secure and Efficient Multiserver Authentication and Key Agreement Protocol for Internet of Vehicles. *IEEE Internet of Things Journa*l, 9(23), 24398-24416
[44] Ying, B.; Nayak, A. Lightweight remote user authentication protocol for multi-server 5G networks using self-certified public key cryptography. *J. Netw. Comput. Appl.* **2019**, *131*, 66–74
[45] Xiong, L.; Li, F.; He, M.; Liu, Z.; Peng, T. An Efficient Privacy-Aware Authentication Scheme With Hierarchical Access Control for Mobile Cloud Computing Services. IEEE Trans. Cloud Comput. 2022, 10, 2309–2323.
[46] Irshad, A., Chaudhry, S. A., Sher, M., Alzahrani, B. A., Kumari, S., Li, X., & Wu, F. (2018). An anonymous and efficient multiserver authenticated key agreement with offline registration centre. IEEE Systems Journal, 13(1), 436-446
[47] Irshad, A., Alzahrani, B. A., Albeshri, A., Alsubhi, K., Nayyar, A., & Chaudhry, S. A. (2023). SPAKE-DC: A Secure PUF Enabled Authenticated Key Exchange for 5G-based Drone Communications. IEEE Transactions on Vehicular Technology
[48] Xie, Q., & Zhao, Y. (2023). Physical-Unclonable-Function-Based Lightweight Three-Factor Authentication for Multiserver Architectures. Mathematics, 12(1), 79.
[49] Das, A. K., Bera, B., Wazid, M., Jamal, S. S., & Park, Y. (2021). On the security of a secure and lightweight authentication scheme for next generation IoT infrastructure. IEEE Access, 9, 71856-71867.
[50] Ghaffar, A., Li, X. J., Ahmad, T., Hussain, N., Alibakhshikenari, M., & Limiti, E. (2020, December). Circularly polarized pattern reconfigurable flexible antenna for 5G-sub-6-GHz applications. In 2020 IEEE Asia-Pacific Microwave Conference (APMC) (pp. 625-627). IEEE

Azeem Irshad received master's degree from Arid Agriculture University, Rawalpindi, Pakistan. Then he completed his PhD from International Islamic University, Islamabad, Pakistan. He has authored more than 90 international journal and conference publications, including 50 SCI-E journal publications. His research work has been cited over 1781 times with 22h-index and 44 i-10-index. He received Top Peer-Reviewer Award from Publons in 2018 with 126 verified reviews. He is serving as Academic editor for SCN and MPE journals (Hindawi). Moreover, he is serving as guest editor for SCN, JHE (Hindawi) and CMC (Techscience)-based special issues. Recently, he has co-edited a book on "IoT and Smart Devices for Sustainable Environment" published by Springer. He has served as a reviewer for more than 40 reputed journals including IEEE Systems Journal, IEEE Communications Magazine, IEEE TII, IEEE Consumer Electronics Magazine, IEEE Sensors Journal, IEEE TVT, IEEE IAS, Computer Networks, Information Sciences, CAEE, Cluster Computing, AIHC, JNCA and FGCS, notably. His research interests include strengthening of authenticated key agreements in Cloud-IoT, smart grid, pervasive edge computing, CPS, 5G networks, WSN, Ad hoc Networks, e-health clouds, SIP and multi-server architectures.

Mohammed Alreshoodi received MSc degree in computer networks from University of Essex, Colchester, UK, in 2011. He also received the Ph.D. degree in in computer networks from University of Essex, Colchester, UK, in 2011. He is currently an Associate Professor in the Applied college at Qassim University, KSA. His research interest includes computer networking, wireless networks, WSN, IoT, networks security. He is a member of WSN research group and Cyber Security research group at Collage of Computer, Qassim University, KSA. He is co-founder and a chairman of the board of Qassim Technology Association.