INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING
DOI: 10.46300/9106.2021.15.11 Volume 15, 2021

A New Remote Fuzzy User Password Authentication Scheme Using Sub-tree for Cloud Computing

Chandrashekhar Meshram[1], Cheng-Chi Lee[2,3*], Muhammad Khurram Khan[4], Kailash Kalare[5], Sarita Gajbhiye Meshram[6]

[1]Department of Post Graduate Studies and Research in Mathematics, Jaywanti Haksar Govt. Post-Graduation College, College of Chhindwara University, Betul, M.P., India.

[2]Department of Library and Information Science Research and Development Center for Physical Education, Health, and Information Technology, Fu Jen Catholic University, New Taipei, Taiwan 24205, R.O.C.

[3]Department of Photonics and Communication Engineering, Asia University, Taichung 41354, Taiwan, R.O.C. *Corresponding

[4]Center of Excellence in Information Assurance, King Saud University, Riyadh, Saudi Arabia

[5]PDPM Indian Institute of Information Technology, Design, and Manufacturing, Jabalpur, Madhya Pradesh, India

[6]Faculty of Environment and Labour Safety, Ton Duc Thang University, Ho Chi Minh City, Vietnam.

Received: June 24, 2020. Revised: January 16, 2021. Accepted: February 6, 2021. Published: February 11, 2021.

Abstract—Recent advancements in internet technology and the infrastructure have attracted more people and organizations to do everything online. Internet technologies have provided amazing and smooth ease for electronic sales and purchases. However, many people have refused to use these internet technologies in electronic purchases because of unstable and insecure forms. New hacking techniques and new types of attacks have been tackled to make these internet technologies better and safer. Smartcard-based password authentication schemes have been the mainstream in recent years, featuring their highly lightweight, easy-to-use equipment and low-cost apps. Various secure and faster authentication schemes have been proposed in the literature. However, most of the existing authentication schemes have found vulnerable to recent attacks and have security flaws. This paper provides and efficient way for authentication using the partial discrete logarithm and sub-tree structure. The proposed scheme has seen effective and more useful in cloud computing environment. The analysis based on the security and the computational cost shows that the proposed authentication scheme proves to be more secure and efficient compared to other protocols that serve the same purposes.

Keywords— Mutual authentication; session key; smartcard; sub-tree; partial discrete logarithm, cloud computing.

I. Introduction

SMARTCARD-based remote user authentication schemes allow a server to authenticate a remote user over public, insecure networks. The systems for authentication typically follow some of the two methods below to identify a user:

- Using something only known to the user, such as a password.
- Using something only the user has legal access to, such as a smart card.

The technology that uses both methods is sometimes referred to as two factor authentications. A smartcard-based password authentication system includes an authentication server AS and a user U. Usually there is three basic phases to the system: registration, login and authentication. However, sometimes an extra phase may also be included for user password change using

the smartcard, usually with the help of AS. Various smartcard-based remote user password authentication schemes have been published with an aim to providing secure and efficient authentication service to online users [2, 5, 8, 17, 18, and 21]. However, most of these schemes are susceptible to some cryptographic attacks. Xu et al. have presented a smartcard-based password authentication scheme [8]. Song et al. [18] and Sood et al. [21] respectively have shown few flaws in Xu et al.'s scheme. Song et al. [18] have shown that the attacker could obtain data from a legitimate user's smartcard and then launch an impersonation attack [12, 15, 24]. Song provided an improved form of Xu et al.'s scheme to fix the problem [17]. Sood et al. [21] shown that Xu et al.'s scheme was weak against the fake attack and the offline password guessing attack. Chen et al. [2] have proven the security faults in both the Song's protocol [17] and Sood et al.'s scheme [21]. Song's scheme is susceptible to the offline password guessing attack and stolen smartcard attack, while Sood et al.'s scheme is incapable of mutual authentication. Then Chen et al. [3] offered additional enhanced smart-card based remote user password authentication scheme. Li et al. [25] have shown that the Chen et al.'s scheme was weak in wrong password detection and failed to provide forward secrecy in the login stage. They additionally claimed that Chen et al.'s password change phase was not user friendly because the user could only change their old password with the help of the server S. H. Islam [19] have identified that Li et al.'s [25] scheme is not only susceptible to the insider attack, known session specific temporary information attack, as well as stolen smartcard attack, but also lacks a mechanism for stolen smartcard revocation. In spite of its impressive achievement on lowering the computation cost, Islam's scheme [19], has seen susceptible to (i) offline password guess attack (ii) stolen smartcard attack(iii) known session specific temporary information attack (iv) as well as user impersonation attack, and it falls short of providing (v) user anonymity and (vi) server's control over the password. Islam's scheme has been analyzed in this work. In this paper, new scheme has been proposed that enables the user to directly replaced password without the help of AS. Moreover, the proposed scheme has a way for the revocation of a lost/stolen smartcard so that a new card can be reissued to the same user. Authentication schemes are essential to all remote operations having to do with online transactions. Therefore, for the whole system to run smoothly, it is important to make sure that these schemes work properly. Burrows–Abadi–Needham (BAN) logic analysis has been used to prove the correctness of the proposed authentication scheme [10]. It has seen that the proposed scheme is performing correctly based on BAN logic. The creation of Internet of Things (IoT) devices is constantly progressing and this growth also causes a number of problems to emerge that increase the complexity of IoT forensic investigation [37] and design aspects of the cloud tier of EMULSION, a generic cloud-based multi-service IoT operating framework built specifically to meet the needs of small and medium-sized companies as a non-expensive IoT platform (SMEs). The EMULSION is a representative of IoT platforms of the new horizontal type, next-generation, that replace the existing vertical type platforms [36].

In this paper, the refinement and update in the Islam's scheme has been done to make the proposed scheme useful in real life situations in modern times. The security check and performance analysis confirm that the proposed scheme has the following merits: (i) the smartcard can detect an incorrect password without having to contact the server;(ii) the user can pick and replace password without server involvement;(iii) the session key is protected against known passive/active attacks;(iv) user and server can commonly authenticate each other and create a typical session key between them (with proof by the BAN logic);(v) compared with earlier schemes, our scheme runs at a lower computation cost and offers more security features.

The organization is as follows. Section 2 provides background information, including brief introductions to the related algorithms and a table of notations to be used throughout this paper. Section 3 provides information about the analysis of the Islam's scheme, followed by a cryptanalysis of the scheme in Section 4. Section 5, give information about the proposed improved scheme. Section 6 provides the analysis of the proposed scheme using the BAN logic. Section 7 gives information about the security analysis and the quantitative comparison based on execution time while the conclusions are given in Section 8.

II. BACKGROUND MATERIAL

This section includes definitions of a couple of algorithms employed by our new scheme, notations and a table of notations to be used throughout this paper.

A. Notations

An authentication scheme for the sharing of data to fuzzy users under the cloud computing environment is a new effort. The notations are the following.

If there is no uncertainty, we use $[\mathcal{v}, \mathcal{w}]$ for the shorthand of $\{\mathcal{v}, \mathcal{v}+1, \ldots, \mathcal{w}\}$ and $[\mathcal{v}]$ for $[1, \mathcal{v}]$. For every $\mathcal{id} = (\mathcal{id}_1, \mathcal{id}_2, \ldots, \mathcal{id}_{\mathcal{k}})$, where $\mathcal{id}$ is an identity vector, let $\mathcal{S}_{\mathcal{id}} = \{\mathcal{id}_1, \ldots, \mathcal{id}_{\mathcal{k}}\}$ be a set of all performing identities in $\mathcal{id}$. $I_{\mathcal{id}} = \{i : \mathcal{id}_i \in \mathcal{S}_{\mathcal{id}}\}$is a location records of $\mathcal{id}$ in the model comprise of tree structure. The predicted receivers form a subtree in an authentication scheme comprise of tree-structured [26, 27, 28]. The $\mathcal{id}$ and the places of their receivers are integrated into $\mathbb{T}$ in the tree structure. The root node must be covered by any legitimate $\mathbb{T}$. This represents the fact that the PKG is managing the structure. Similarly, $\mathbb{T}$'s identity set and $\mathbb{T}$'s location indices are represented by $\mathcal{S}_{\mathbb{T}} = \cup_{\mathcal{id} \in \mathbb{T}} \mathcal{S}_{\mathcal{id}}$ and $I_{\mathcal{id}} = \{i : \mathcal{id}_i \in \mathcal{S}_{\mathbb{T}}\}$. The symbolisations here can be expressed as $Sup\ (\mathcal{id}) = \{(\mathcal{id}_1, \mathcal{id}_2, \ldots, \mathcal{id}_{\mathcal{k}'}) : \mathcal{k}' \leq \mathcal{k}\}$ to indicate the superiority of $\mathcal{id} = \mathcal{id}_1, \mathcal{id}_2, \ldots, \mathcal{id}_{\mathcal{k}})$. Subtree $\mathbb{T}$'s projected receivers are characterized as $Sup\ (\mathbb{T}) = \cup_{\mathcal{id} \in \mathbb{T}}\ Sup\ (\mathcal{id})$.

The symbolisations fit for our proposed scheme is based on sub-tree have been discussed here. Suppose that users are structured as in a tree structure. The respective identity set $\mathcal{S}_{\mathcal{id}} = \{\mathbb{B}, \mathbb{F}\}$ and position indices $I_{\mathcal{id}} = \{2, 6\}$ of the, to specify a predetermined user with $\mathcal{id} = (\mathbb{B}, \mathbb{F})$. The user creates a set of $Sup\ (\mathcal{id}) = \{(\mathbb{B}), (\mathbb{B}, \mathbb{F})\}$ which involves himself/him and herself/her superiors. When an data owner sends a message in a subtree consist of set of receivers such as $\mathbb{T} = \{(\mathbb{A})(\mathbb{B}, \mathbb{F}), (\mathbb{B}, \mathbb{G})\}$, we are denoting $\mathbb{T}$'s identity set and position indices as $S_{\mathbb{T}} = \{\mathbb{A}, \mathbb{B}, \mathbb{F}, \mathbb{G}\}$, and $I_{\mathbb{T}} = \{1, 2, 6, 7\}$ respectively.

$\mathbb{T}$'s superiors are described as $Sup(\mathbb{T}) = \{(\mathbb{A}), (\mathbb{B}), (\mathbb{B}, \mathbb{F}), (\mathbb{B}, \mathbb{G})\}$, which is clearly the user agreement that the owner of the data wants to convey.

Definition 1: Discrete logarithm: Given two numbers $(g, g^{\alpha} \bmod p)$, finding α within polynomial time is difficult, where $\alpha \in Z_p^*$ and 'g' is the primitive component of the cyclic group Z_p^*.

Definition 2: Diffie Hellman algorithm: Particular values of $(g, g^{\alpha} mod\ p, g^{\beta} mod\ p)$, finding $(g^{\alpha\beta} mod\ p)$ within polynomial time is difficult, where$\alpha, \beta \in Z_p^*$.

Definition 3: Partial Discrete Logarithm [29, 30]:Let $g \in \mathcal{G} = QR_{n^2}$ with maximal order, for straightforwardness, we suppose that $g^{\lambda(n)} mod\ n^2 = (n+1)\ mod\ n^2$, that is $k = 1$. For given g and $z = g^{a} mod\ n^2$ (for $a \in [1, ord\ (\mathcal{G})]$), Paillier [29] characterized the *Partial Discrete Logarithm (PDL)* as the computational issue of registering $a\ (mod\ n)$.

Definition 4: Partial Discrete Logarithm over$\mathcal{Z}_{n^2}^*$[32, 31]: For each probabilistic polynomial time (PPT) algorithm A, there exist a negligible function $negl()$with the end goal for sufficiently large v.

$$\Pr[A(n, g, z) = a\ mod\ n \mid q, p \leftarrow SP(v/2); n = qp; g \leftarrow \mathcal{G}; a \leftarrow [1, ord\ (G)]; z \leftarrow g^{a} mod\ n^2] = negl(v)$$

Table I. Notations to be used in both Islam's scheme and our new scheme

Notation	Meaning
U_i	The i^{th} user.
id_i	Exclusive identification ofU_i
sid_i	Exclusive identification of U_i associated with subtree $\mathbb{T}$
PW_i	Exclusive password of U_i.
AS	The Authentication Server.
SC	Smartcard.
α, β	Session-specific temporary values generated by user and server respectively.
n	Large integer to be a sheltered prime modulus, where$n = p.q$ and $= 2p' + 1$,$q = 2q' + 1$, in which p, q, q' and p' are safe primes.
ΔT	Maximum transmission delay.
$h(.)$	One-way hash function.
sid_i	UserU_i's smartcard id.
$ssid_i$	UserU_i's smartcard idassociated with subtree $\mathbb{T}$
X	Private key maintained by server.
$\oplus$	The Exclusive-OR operation.
$\|\|$	The string concatenation operation.
SK	Session key.

III. REVIEW OF ISLAM'S SCHEME [19]

This section gives information about the smartcard-based authentication scheme and it's working. The full protocol of Islam's scheme has been illustrated by Figure 1 in the form of a workflow. At first, the server AS randomly picks two vast primes q and p, a private key x, and a protected, lightweight one-way hash function $h(.)$consisting of only:$\{0,1\}^*$.

A. Registration Phase

R1:$U_i$picks her/his identity id_i and drives it to AS in a safe manner.

R2: AS verifies the validity of id_i. If it is invalid, AS approaches$U_i$for a new identity. Otherwise, AS selects a brand fresh SC, obtains its identity sid_i, and processes $C_i = h(id_i||x||sid_i)$.

R3: AS stores$\{id_i;\ sid_i\}$ about $U_i$in its database.

R4: AS writes $\{C_i;\ q;\ p;\ h(.)\}$ into the SC and directsthe message to $U_i$over a safe channel.

R5:Upon receiving the SC,$U_i$encloses his/her password PW_i into the SC by computing: $B_i = C_i \oplus h(PW_i) = h(id_i||x||sid_i) \oplus h(PW_i)$ and $A_i = C_i^{PW_i}(mod\ p) = h(id_i||x||sid_i)^{PW_i}\ (mod\ p)$.

R6:The SC replaces C_i with B_i and stores A_i.Now, the SC contains the messages $\{A_i;\ B_i;\ p;\ q;\ h(.)\}$.

B. Login Phase

L1: The user $U_i$embeds her/his SC into a smartcard peruser and after thatentersher/his identityid_i and password PW_i. Then the SC accomplishes the accompanying calculations.

L2: The SC computes $C_i = B_i \oplus h(PW_i)$ and $A_i^* = C_i^{PW_i}(mod\ p)$.

L3: The SC verifies $(A_i^* =?\ A_i)$. If the verification turns out negative, the SC denies U_i 's login appeal; else, the SC goes to the following step.

L4: The SC picks the current timestamp $T_i$as well as the session-specific random number α, and computes $D_i = C_i^{\alpha}(mod\ p) = h(id_i||x||sid_i)^{\alpha}\ (mod\ p)$, $M_i = h(id_i||C_i||D_i||T_i)$.

L5: The SC sends U_i's login message $\{id_i;\ D_i;\ M_i;\ T_i\}$ to AS over a public channel at time T_i.

C. Authentication Phase

After getting U_i's message for login appeal at time T_i', the server AS completes the accompanying steps:

A1: AS verifies the legitimacy of$id_i$and confirms the legitimacy of the timestamp by checking if $T_i' - T_i <= \Delta T$. If either of the verifications does not check out, AS denies U_i's login request; otherwise, AS goes on with the following steps.

A2: AS Calculates $C_i^1 = h(id_i||x||sid_i)$ and $M_i^* = h(id_i||C_i^1||D_i||T_i)$ and then retrieves sid_i from the database and checks $M_i^* =?\ M_i$for user authentication.

A3: For SK generation, AS chooses $\beta\ \epsilon_R Z_P^*$andcomputes $V_i = C_i^{\beta}\ (mod\ p)\ =\ h(id_i||x||sid_i)^{\beta}\ (mod\ p)$ and $SK = D_i^{\beta}\ (mod\ p)\ =\ h(id_i||x||sid_i)^{\alpha\beta}\ (mod\ p)$.

Registration Stage

U_i (User) / smartcard **Authentication Server (AS)**

U_i selects id_i $\{id_i\}$ Via secure channel → Check id_ivalidity, if it is not valid, ask for new valid id_i
Else selects smartcard identity sid_i
Compute: $C_i = h(id_i||x||sid_i)$,
Store $\{id_i, sid_i\}$ in its database

$S.C = \{C_i, p, q, h(.)\}$ along wth sid_ivia secure channel ←

Insert PW_iinto the smartcard

Smartcard:

Computes: $B_i = C_i \oplus h(PW_i) = h(id_i||x||sid_i) \oplus h(PW_i)$

And$A_i = C_i^{PW_i}(mod\ p)\ =\ h(id_i||x||sid_i)^{PW_i}\ (mod\ p)$.

ReplaceC_i by B_i and store A_i

Finally, Smartcard (SC) contains $\{A_i;\ B_i;\ p;\ q;\ h(.)\}$

Login Stage

U_i Submits$\{id_i,\ PW_i\}$

SC Compute:

$C_i = B_i \oplus h(PW_i)$ and $A_i^* = C_i^{PW_i}(modp)$.

If $(A_i^*\ =?\ A_i)$, ask for new $\{id_i,\ PW_i\}$

Else choose $\alpha\ \epsilon_R Z_P^*$, and

$D_i = C_i^{\alpha}(mod\ p)\ =\ h(id_i||x||sid_i)^{\alpha}\ (mod\ p)$,

And $M_i = h(id_i||C_i||D_i||T_i)$

$\{id_i\ ;\ D_i\ ;\ M_i;\ T_i\}$ via public channel →

Authentication Stage

Verifyid_i and $T_i' - T_i <= \Delta T$,
If it is not valid, rejects the request.
Else Compute:
$C_i^1 = h(id_i||x||sid_i)$and $M_i^* = h(id_i||C_i^1||D_i||T_i)$,
And verifies $M_i^*\ =?\ M_i$, rejects the request
Else choose $\beta\epsilon_R Z_P^*$, and compute:
$V_i = C_i^{\beta}\ mod\ p = h(ID_i||x|sid_i)^{\beta}\ (mod\ p)$ and
$SK = D_i^{\beta} mod\ p = h(id_i||x||sid_i)^{\alpha\beta}\ (mod\ p)$.
$M_s = h(id_i||C_i^1||V_i||SK||T_s)$

$\{id_i;\ V_i;\ M_s;\ T_s\}$via public chanel ←

Check id_i and Verify: $(T_s^1 - T_s) <= \Delta t$

If either of them is invalid, then rejects the session.

Else compute: $SK^* = V_i^{\alpha}(mod\ p) = h(id_i||x||sid_i)^{\alpha\beta}\ (mod\ p)$

and $M_s^* = h(id_i||C_i^1||V_i||SK^*||T_s))$

If $(M_s^* \neq M_s)$, rejects the session

Else accept *SK* and S is authenticated.

Password Change Stage

U_iSubmits $\{id_i;\ PW_i\}$

Compute:

$C_i = B_i \oplus h(PW_i)\ = h(id_i||x||sid_i)$,

And $A_i^* = C_i^{PW_i} (mod\ p)$
If $A_i^* \neq A_i$, rejects the request
Else SC computes: $B_i^{new} = B_i \oplus h(PW_I) \oplus h(PW_i^{new})$
$A_i^{new} = C_i^{PW_i^{new}} (mod\ p)$
replaces $(A_i, B_i)$with (A_i^{new}, B_i^{new}),

Smartcard Revocation Stage

$\{id_i\}$ via secured channel →

Checkid_ivalidity using personal information
Selects new smartcard with identity sid_i^*
Compute: $C_i^* = h(id_i||x||sid_i^*)$,

$S.C = \{ C_i^*, p, q, h(.) \}$ along wth sid_i^*via secure channel ←

Insert new PW_i^* into the smartcard
SC Computes: $B_i^* = C_i^* \oplus h(PW_i^*) = h(id_i||x||sid_i^*) \oplus h(PW_i^*)$
and $A_i^* = (C_i^*)^{PW_i^*} (mod\ p) = h(id_i||x|sid_i^*)^{PW_i^*} (mod\ p)$.
Replace C_i^* by $B_i^*$and store A_i^*
Finally, $S.C$ contains $\{ A_i^*, B_i^*, p, q, h(.)\}$

Figure 1. Islam's scheme [19]

A4: AS marks the recent timestamp T_s, processes $M_s = h(id_i||C_i^1||V_i||SK||T_s)$, and sends the response message $\{id_i;\ V_i;\ M_s;\ T_s\}$ to $U_i$over a public channel.
A5: On accepting the login answer message at time T_s^1, the user $U_i$continues to verify thelegitimacy of id_i andthe legitimacy of the time interim amongst T_s^1 and T_s. If either of the above verifications fails, U_i terminates the session. Otherwise, U_i proceeds as follows.
A6: U_i computes
$SK^* = V_i^{\alpha} (mod\ p) = h\left(id_i||x||sid_i\right)^{\alpha\beta} (mod\ p)$ and $M_s^* = h(id_i||C_i^1||V_i||SK^*||T_s))$ and then verifies $(M_s^* =?\ M_s)$ for server authentication.
Note: A common session key $SK = h(id_i||x||sid_i)^{\alpha\beta}$ is shared between U_i and AS for subsequent communications. The SK stays fresh only during the current session and will be modified for subsequent sessions.

A. Password Change Phase

In this stage, the user can modify hers/his old password PW_i to a fresh password PW^{new} without the assistance of AS.The steps as per the following:
P1: U_i embeds his/her SC into a card per user, enters the old $\{id_i;\ PW_i\}$, and sends out a password change request.
P2: After accepting the password change appeal from U_i, AS calculates $C_i = B_i \oplus h(PW_i) = h(id_i||x||sid_i)$, $A_i^* = C_i^{PW_i} (mod\ p)$. Then SC verifies $A_i^* =?\ A_i$. If the verification turns out negative, AS either terminates the sequence or asks $U_i$to enter a new password PW_i^{new}.
P3: SC computes $B_i^{new} = B_i \oplus h(PW_I) \oplus h(PW_i^{new})$ and $A_i^{new} = C_i^{PW^{new}} (mod\ p)$.
P4: The SCreplaces(A_i, B_i) with (A_i^{new}, B_i^{new}). Finally, the SC contains the information $\{ A_i^{new}, B_i^{new}, p, q, h(.) \}$.

B. Smartcard Revocation Phase

To provide more flexibility and higher-level security to the user, the cancellation of a lost/stolen *SC* with a new one reissued with same login identity [14] is one of the basic necessities a smartcard-based authentication scheme should satisfy. The steps are as follows:
V1: $U_i$sends his/her old id_i along with id proofs (PAN number, Voter card number or date of birth) to *AS*.
V2: *AS* disputes a fresh *SC*, gets its identity sid_i^*, and processes $C_i^* = h(id_i||x||\ sid_i^*)$. Then AS writes $\{C_i^*;\ q;\ p;\ h(.)\}$ into the *SC* and sends the new *SC* to$U_i$through a safe channel. *AS* stores$\{id_i;\ sid_i^*\}$ against U_i in its database.
V3:Upon receiving the *SC*, $U_i$places her/his fresh password PW_i into the *SC* and after that *SC* calculates $B_i = C_i \oplus h(PW_i) = h(id_i||x||sid_i) \oplus h(PW_i)$ and $A_i = C_i^{PW_i} (mod\ p) = h\left(id_i||x||sid_i\right)^{PW_i} (mod\ p)$.
V4: The *SC* substitutes B_i forC_i and stores A_i. Now, the *SC* contains the information $\{A_i;\ B_i;\ h(.);\ p;\ q\}$.

IV. CRYPTANALYSIS OF THE ISLAM SCHEME

This section gives information about some security weaknesses in the Islam's scheme. Each of the subsections will focus on one security flaw.

A. Failure to Withstand Smart Card Breach Attack

Numerous scholars have demonstrated that the information kept in the SC can be taken out by utilizing different strategies like power investigation and so on [1, 4, 10, 11, 12, 15, 22, 24]. That is to say, an attacker *E* can separate the information $\{A_i;\ B_i;\ p;\ q;\ h(.)\}$written in some SC of a legal user U_i if the attackersomehow gets to hold that *SC* for a period of time. (The attacker can record those values and then return the *SC*.) The attacker *E* can guess the user's password PW_i with the

knowledge of A_i and B_i extracted from the *SC*, as explained below.

B. Vulnerability to Off-line Password Guessing Attack

As we mentioned above, the attacker 'E' has a way to obtain A_i and B_i from the *SC*, where $A_i = C_i^{PW}(mod\ p)$ and $B_i = C_i \oplus h(PW_i)$. Expression B_i can also be rewritten as $C_i = B_i \oplus h(PW_i)$. Hence, the attacker E can frame $A_i = \left(B_i \oplus h(PW_i)\right)^{PW_i}(mod\ p)$. Since the attacker knows A_i, B_i, $h(.)$, and p but PW_i, there is room to perform the off-line password guess attack [20] repeatedly until the correct password is reached. Then, E can also obtain id_i directly by snooping the communication messages (e.g., login message $\{id_i;\ D_i;\ M_i;\ T_i\}$, login reply message $\{id_i;\ V_i;\ M_s;\ T_s\}$) traded amongst U_i and AS, as the messages are traded through a public uncertain correspondence channel, namely the Internet. Now that the attacker knows id_i and PW_i of U_i, she/heis ready to dispatch a wide range of attacks.

C. Vulnerability to Known Session-Specific Temporary Information Attack

To withstand this attack, all session keys must be under proper protection, making sure that no harm can be done even if the session-specific arbitrary numbers are known to an attacker *E*. As per [6, 16, and 33], safeguard against this attack is vital, and this kind of attack is most likely to take effect for the following reasons [13, 23]:

(1) The server and user rely on a random number generator of some external or internal source that maybe compromised by the attacker *E*.

(2) During each communication session, the random numbers are stored in the device. If they are not deleted immediately when the session terminates, then the attacker *E* might have access to them by taking control of the server's or the user's device.

In Islam's scheme, U_i and AS compute the session key $SK = D_i^{\beta}(mod\ p)(or)\ V_i^{\alpha}(mod\ p)$, where α and β are the session-specific arbitrary records picked by U_i and *AS* individually. The attacker E can then directly acquire D_i, V_i by eavesdropping messages (login request and reply messages) transmitting through insecure public communication channels. The session key *SK* can be easily cooperated if α and β are known to *E*. Hence, Islam's scheme is vulnerable to the known session-specific temporary information attack.

D. Failure to Preserve User Anonymity

In Islam's scheme, what the user U_i sends to the server *AS* as the login appeal message is $\{id_i;\ D_i;\ M_i;\ T_i\}$, and what *AS* sends back to the user U_i as the login reply message is $\{id_i;\ V_i;\ M_s;\ T_s\}$. Observing the id_i above messages, the attacker *E* comes to speculate that the same part id_i that appears in both messages may have something to do with an user's identity. Using this id_i, the attacker can then derive the other values (e.g., sid_i) if he/she can compromise the server's database, since the server keeps $\{id_i, sid_i\}$ for each user U_i. This is why we say Islam's scheme neglects to save user secrecy.

E. Failure to Resist User Impersonation Attack

This kind of attack happens when an attacker pretends to be a valid user and forges the authentication message using some information acquired from the authentication protocol. The attacker can attempt to alter a login demand message $\{id_i, D_i, M_i, T_i\}$ into $\{id_i, D_i^*, M_i^*, T_i^*\}$ so as to succeed in the authentication phase, where T_i^* is the point of time when the login message is sent out. $D_i^* = C_i^{\alpha}\ mod\ p$, $M_i^* = h(id_i||C_i||D_i||T_i)$. The attacker can then obtain: C_i by launching a password guessing attack as discussed in Subsection 4.2; p from the smartcard as discussed in Subsection 4.1; α simply by picking any random number; id_i by monitoring previous login request messages. Consequently, the attacker can effectively get a login demand message through, and the server will respond to it by sending a login reply message $\{id_i, V_i, M_s, T_s\}$. Now the attacker can frame $SK = (V_i^{\alpha} mod\ p)$ and proceed to do further communication with the server. This means the scheme is vulnerable to the user impersonation attack.

V. THE PROPOSED SCHEME

This section gives information about the proposed password authentication scheme presented in this paper. The representations used in our scheme are the same as those in Islam's scheme. The proposed scheme also comprises of five stages as listed below. Figure 2 is the workflow of the proposed scheme. The steps of each of the five phases are discussed below.

A. Registration Phase

This stage is a single-time performance phase that happens when a user U_i registers with the remote server *AS*. The progressions to take are as follows:

R1: The user U_i first choosesan identity sid_i associated with subtree $sup\mathbb{T}$ and a safe password PW_i. Then he/she computes $RPW_i = h(b_i \oplus PW_i)$, where b_i is an arbitrary numeric value.

R2: $U \rightarrow AS$: $\{sid_i, RPW_i\}$ through a safe correspondence channel.

R3: On accepting the enrollment demand from U_i at time T_i, *AS* continues to check whether sid_i exists or not.In the event that it exists, *AS* rejects the enrollment demand; otherwise it continues to produce a *SC* identity $ssid_i$ specific to U_i and compute $C_i = h(sid_i||x||ssid_i)$, $Tsid_i = h(T_i||x)$,and $SD_i = ssid_i \oplus Tsid_i$.

Note that *AS* stores $\{Tsid_i, h(sid_i), SD_i\}$ for each user U_i in its database, where x is a server secrete key.

R4: $AS \rightarrow U_i$, a *SC* containing $\{C_i, n, h(.)\}$ along wth T_i and $ssid_i$ to the user U_i by means of a secured correspondence channel.

R5: On receiving the *SC*, U_i computes B_i, A_i, R_i, S_i and writes these values into the *SC*. Now the *SC* contains $\{A_i, B_i, R_i, S_i, n, h(.)\}$, where $B_i = C_i \oplus h(PW_i||b_i||sid_i)$, $A_i = h(b_i||sid_i||PW_i||C_i)$, $R_i = b_i \oplus h(PW_i||sid_i)$, $S_i = T_i \oplus h(sid_i||b_i||PW_i)$.

B. Login Phase

To receive service from the AS, a user U_i has to embed her/his SC into the card peruse and submit her/his sid_i and PW_i.At that point, the SC executes the accompanying phases.

L1: Compute: $b_i = R_i \oplus h(PW_i||sid_i)$, $C_i = B_i \oplus h(PW_i||b_i||sid_i)$, $A_i^* = h(b_i||sid_i||PW_i||C_i)$.
L2: Compare the computed A_i^* with A_i, which is put away in U_i's smart card *SC*. If both are in distinguishable, the validity of the user is accepted, and the *SC* goes on to take the next step.
L3: Pick a session-specific arbitrary number α, and calculate: $D_i = C_i^{(\alpha b_i)} (mod\ n^2)$, $T_i = S_i \oplus h(sid_i||b_i||PW_i)$, $M_i = h(sid_i||C_i||D_i||\ T_i\ ||\ T_1)$, where T_1 is the current time, $Dsid_i = id_i \oplus h(ssid_i\ ||\ T_1\ ||\ T_i)$.
L4: *SC* sends the login request message $\{Dsid_i, D_i, M_i, T_i, T_1\}$ to *AS*.

U_i		AS
Registration Stage		
$U_i$selects sid_i, PW_i, b_i		
Computes $RPW_i = h(b_i\|\|PW_i)$	$\{sid_i, RPW_i\}$ →	
		Received at Time T_i
		Check sid_ivalidity, if it is not valid, ask for new valid sid_i
		Else selects smartcard identity $ssid_i$
		Compute: $C_i = h(sid_i\|\|x\|\|ssid_i)$, $Tsid_i = h(T_i\|\|x)$ and $SD_i = ssid_i \oplus Tsid_i$.
		Store $\{Tsid_i, h(sid_i), SD_i\}$ in its database
	← $SC = \{C_i, n, h(.)\}$ along wth T_i and $ssid_i$	
Computes: $B_i = C_i \oplus h(PW_i\|\|b_i\|\|sid_i)$, $A_i = h(b_i\|\|sid_i\|\|PW_i\|\|C_i)$, $R_i = b_i \oplus h(PW_i\|\|sid_i)$, $S_i = T_i \oplus h(sid_i\|\|b_i\|\|PW_i)$		
Inserts B_i, A_i, R_i, S_i values into the *SC*, in place of C_i		
Finally, $S.C$ contains $\{B_i, A_i, R_i, S_i, n, h(.)\}$		
Login Stage		
U_iSubmits $\{sid_i, PW_i\}$		
Compute: $b_i = R_i \oplus h(PW_i\|\|sid_i)$, $C_i = B_i \oplus h(PW_i\|\|b_i\|\|sid_i)$, $A_i^* = h(b_i\|\|sid_i\|\|PW_i\|\|C_i)$		
If $(A_i^* \neq A_i)$, ask for new $\{sid_i, PW_i\}$		
Else choose a session specific random number α, and Compute: $D_i = C_i^{(\alpha b_i)} (mod\ n^2)$, $T_i = S_i \oplus h(sid_i\|\|b_i\|\|PW_i)$, $M_i = h(sid_i\|\|C_i\|\|D_i\|\|\ T_i\ \|\ T_1)$, $Dsid_i = sid_i \oplus h(ssid_i\ \|\ T_1\ \|\ T_i)$		
	$\{Dsid_i, D_i, M_i, T_i, T_1\}$via public channel →	
Authentication Stage		
		Verifies:$(T_1' - T_1) <= \Delta t$,
		If it is not validity, rejects the request.
		Else Compute: $Tsid_i = h(x\|\|T_i)$.
		$ssid_i^* = SD_i \oplus Tsid_i$.
		$sid_i^* = Dsid_i \oplus h(ssid_i^*\ \|\ T_1\|\ T_i)$.
		And verifies $h(sid_i^*) =? h(sid_i)$.
		If it is not validity, then rejects the request.
		Else choose session specific random number β, and compute: $C_i^* = h(sid_i\|\|x\|\|ssid_i)$
		$M_i^* = h(sid_i\|\|C_i^*\|\|D_i\|\|\ T_i\|\|\ T_1)$
		$V_i = (C_i^*)^{\beta s.id_i} (mod\ n^2)$
		$SK = (D_i)^{\beta s.id_i} (mod\ n^2) = (C_i)^{\alpha b_i \beta sid_i}$

$M_s = h(\text{s}id_i || C_i^* || V_i || SK || T_2)$
$\text{s}\mathcal{s}Did_i = D\text{s}id_i \oplus h(C_i || T_2)$
$T_i^* = T_i \oplus h(T_1 || \beta)$
$T_i^{new} = T_i^* \oplus h(T_2 || \text{s}id_i)$.
AS updates $SD_i = \mathcal{s}\text{s}id_i \oplus h(x || T_i^*)$in itsDB

$\{\mathcal{s}D\text{s}id_i;\ V_i;\ M_s;\ T_2;\ T_i^{new}\}$via public channel ←

Verify: $(T_2' - T_2) <= \Delta t$
If it is not validity, then rejects the request.
Else compute:
$D\text{s}id_i^* = \mathcal{s}D\text{s}id_i \oplus h(C_i || T_2)$
$\text{s}id_i^* = D\text{s}id_i^* \oplus h(\mathcal{s}\text{s}id_i || T_1 || T_i)$
Verifies $id_i^* =? id_i$ for id verification.
$SK^* = V_i^{(\alpha . b_i)} (mod\ n^2) = (C_i)^{\alpha b_i \beta \text{s}id_i} (mod\ n^2)$
$M_s^* = h(\text{s}id_i || C_i || V_i || SK^* || T_2)$
If $(M_s^* \neq M_s)$ rejects the request.
Else, compute:
$T_i^* = T_i^{new} \oplus h(T_2 || \text{s}id_i)$
$S_i^* = T_i^* \oplus h(\text{s}id_i || b_i || PW_i)$
Updates new Si* in its smartcard

→ {Actual Message, $T_2 + 1\}_{SK}$ via public channel

Password Change Stage
U_iSubmits$\{\text{s}id_i, PW_i\}$
Compute:
$b_i = R_i \oplus h(PW_i || \text{s}id_i)$,
$C_i = B_i \oplus h(PW_i || b_i || \text{s}id_i)$,
$A_i^* = h(b_i || \text{s}id_i || PW_i || C_i)$
If $(A_i^* \neq A_i)$, ask for new $\{\text{s}id_i, PW_i\}$
ElseU_i entersPW_i^{new}, Then *SC* computes:
$T_i = S_i \oplus h(\text{s}id_i || b_i || PW_i^{old})$,
$B_i^* = C_i \oplus h(PW_i^{new} || b_i || \text{s}id_i)$,
$A_i^* = h(b_i || \text{s}id_i || PW_i^{new} || C_i)$,
$R_i^* = b_i \oplus h(PW_i^{new} || \text{s}id_i)$,
$S_i^* = T_i \oplus h(\text{s}id_i || b_i || PW_i^{new})$.
Inserts $B_i^*, A_i^*, R_i^*, S_i^*$values into the *SC*
Finally, *S.C* contains $\{h(.), n, B_i^*, A_i^*, R_i^*, S_i^*\}$

Smartcard Revocation Stage

$\{\text{s}id_i\}$ via secured channel →

Received at Time T_i^*
Checks id_ivalidity using personal information
Selects new smartcard with identity $\mathcal{s}\text{s}id_i^*$
Compute: $C_i^* = h(\text{s}id_i || x || \mathcal{s}\text{s}id_i^*)$,
$T\text{s}id_i^* = h(x || T_i^*)$and $SD_i^* = \mathcal{s}\text{s}id_i^* \oplus T\text{s}id_i^*$.
Store $\{T\text{s}id_i^*, h(\text{s}id_i), SD_i^*\}$in its database

$S.C = \{h(.), n, C_i^*\}$ along wth T_i^* and $\mathcal{s}\text{s}id_i^*$via secure channel ←

Computes: $B_i^* = C_i^* \oplus h(PW_i || b_i || \text{s}id_i)$,
$A_i^* = h(b_i || \text{s}id_i || PW_i || C_i^*)$,
$R_i = b_i \oplus h(PW_i || \text{s}id_i)$,
$S_i = T_i^* \oplus h(\text{s}id_i || b_i || PW_i)$
Inserts B_i^*, A_i^*, R_i, S_i^*values into the *SC*, in place of C_i^*
Finally, *S.C* contains $\{h(.), n, B_i^*, A_i^*, Ri, S_i^*\}$

Figure 2. Our proposed scheme

A. Authentication Phase

Upon accepting the login demand message from $U_i$at timeT_1', *AS* completes the ensuing undertakings:

A1: Test the legitimacy of the timestamp by checking if $(T_1' - T_1) <= \Delta t$. If the timestamp checks out, then *AS* takes the following steps.

A2: To authenticate U_i, AS computes: $T\mathrm{s}id_i = h(x||T_i)$.

$\mathcal{s}\mathrm{s}id_i^* = SD_i \oplus T\mathrm{s}id_i$, $\mathrm{s}id_i^* = D\mathrm{s}id_i \oplus h(\mathcal{s}\mathrm{s}id_i^*||T_1||T_i)$. And verifies $h(\mathrm{s}id_i^*) =? h(\mathrm{s}id_i)$.If the above verifications come out positive, then U_i is a legitimate user; otherwise, the login request is terminated immediately.

A3: *AS* chooses a session-specific arbitrary number β and does the accompanying calculations:

$C_i^* = h(\mathrm{s}id_i||x||\mathcal{s}\mathrm{s}id_i)$

$M_i^* = h(\mathrm{s}id_i||C_i^*||D_i||T_i||T_1)$

$V_i = (C_i^*)^{\beta.\mathrm{s}.id_i}(mod\ n^2)$

$SK = (D_i)^{\beta.\mathrm{s}id_i}(mod\ n^2) = (C_i)^{\alpha b_i \beta \mathrm{s}id_i}(mod\ n^2)$

$M_s = h(\mathrm{s}id_i||C_i^*||V_i||SK||T_2)$

$\mathrm{s}\mathcal{s}Did_i = D\mathrm{s}id_i \oplus h(C_i||T_2)$

$T_i^* = T_i \oplus h(T_1||\beta)$

$T_i^{new} = T_i^* \oplus h(T_2||\mathrm{s}id_i)$.

AS modifies $SD_i = \mathcal{s}\mathrm{s}id_i \oplus h(x||T_i^*)$ in its database in response to the new timestampT_i^*.

A4: $AS \rightarrow U_i$ transmits a login reply message $\{\mathcal{s}D\mathrm{s}id_i;\ V_i;\ M_s;\ T_2;\ T_i^{new}\}$ at time T_2.

A5: Upon getting the login demand reply message, U_i checks time legitimacy $(T_2' - T_2) <= \Delta t$.

A6: If the time interval checks out, then it calculates $D\mathrm{s}id_i^* = \mathcal{s}D\mathrm{s}id_i \oplus h(C_i||T_2)$, $\mathrm{s}id_i^* = D\mathrm{s}id_i^* \oplus h(\mathcal{s}\mathrm{s}id_i||T_1||T_i)$. U_i verifies $id_i^* =? \mathrm{s}id_i$ for id verification and calculates $SK^* = V_i^{(\alpha.b_i)}(mod\ n^2) = (C_i)^{\alpha b_i \beta \mathrm{s}id_i}(mod\ n^2)$, $M_s^* = h(\mathrm{s}id_i||C_i||V_i||SK^*||T_2)$.

A7: $U_i$contrasts the calculatedM_s^*esteem and the got $M_s$value. In the event that they match, server authentication is completed; otherwise, this session is terminated immediately.

A8: U_i can calculate $T_i^* = T_i^{new} \oplus h(T_2||\mathrm{s}id_i)$, as U_i can provide his/her own$\mathrm{s}id_i$ and T_2. Then $U_i$changes $S_i^* = T_i^* \oplus h(\mathrm{s}id_i||b_i||PW_i)$in response to the new timestampT_i^* in his/her *SC*.

A9: All further communications are under encryption with the framed session key *SK* between U_i and AS.

B. Password Change Phase

In this stage, U_i can alter her/his old password PW_i to a fresh password PW_i^{new} without the help of the server. The steps are as follows:

P1: U_i insertsher/his *SC* into a card reader, enters her/his old $(\mathrm{s}id_i, PW_i)$, and opts for a password change request.

P2: The *SC* computes $b_i = R_i \oplus h(PW_i||\mathrm{s}id_i)$, $C_i = B_i \oplus h(PW_i||b_i||\mathrm{s}id_i)$ and $A_i^* = h(b_i||\mathrm{s}id_i||PW_i||C_i)$.

P3: The *SC* verifies $A_i^* =? A_i$. If the two do not match, then the *SC* deniesU_i's request; else, the user is allowed to choose a fresh passwordPW_i^{new}.

P4: The *SC* computes $T_i = S_i \oplus h(\mathrm{s}id_i||b_i||PW_i^{old})$, $B_i^* = C_i \oplus h(PW_i^{new}||b_i||\mathrm{s}id_i)$, $A_i^* = h(b_i||\mathrm{s}id_i||PW_i^{new}||C_i)$, $R_i^* = b_i \oplus h(PW_i^{new}||\mathrm{s}id_i)$, $S_i^* = T_i \oplus h(\mathrm{s}id_i||b_i||PW_i^{new})$, and then the new values $B_i^*, A_i^*, R_i^*, S_i^*$are written into the *SC*. Now the *SC* contains $\{h(.), n, B_i^*, A_i^*, R_i^*, S_i^*\}$, and U_i can login with PW_i^{new}.

C. Stolen SC Revocation Phase

In our scheme, if the user should lose his/her SC, then he/she can cancel the lost card and get a one reissued with the same login $\mathrm{s}id$.The steps are as follows:

R1: U_i sends his/her old $\mathrm{s}id_i$ to AS.

R2: *AS* checks U_i's $\mathrm{s}id_i$ and other personal information (e.g., Aadhaar card, voter card, PAN card, or date of birth, etc.) from which$U_i$can be uniquely renowned.

R3: *AS* issues a new *SC* with the identity being $\mathcal{s}\mathrm{s}id_i^*$ and computes: $C_i^* = h(\mathrm{s}id_i||x||\mathcal{s}\mathrm{s}id_i^*)$, $T\mathrm{s}id_i^* = h(x||T_i^*)$ and $SD_i^* = \mathcal{s}\mathrm{s}id_i^* \oplus T\mathrm{s}id_i^*$.

R4: *AS* stores $\{T\mathrm{s}id_i^*, h(\mathrm{s}id_i), SD_i^*\}$ in its database.

R5: $AS \rightarrow U_i$, dispatch a *SC* containing $\{h(.), n, C_i^*\}$ along with T_i^* and $\mathcal{s}\mathrm{s}id_i^*$via a protected channel.

R6: On accepting the SC, U_i computes $B_i^* = C_i^* \oplus h(PW_i||b_i||\mathrm{s}id_i)$, $T_i^* = h(b_i||\mathrm{s}id_i||PW_i||C_i^*)$, $R_i = b_i \oplus h(PW_i||\mathrm{s}id_i)$, $S_i = T_i^* \oplus h(\mathrm{s}id_i||b_i||PW_i)$

R7: SubstituteB_i^*, A_i^*, R_i, S_i^* into the *SC* in place of C_i^*. Now the SC contains $\{h(.), n, B_i^*, A_i^*, Ri, S_i^*\}$.

VI. FORMAL AUTHENTICATION PROOF BASED ON BAN LOGIC

Mutual authentication and session key establishment are the most crucial parts of an authentication scheme and are thus the most important parts to evaluate when it comes to deciding which scheme is the optimal choice. Burrows et al. [10] proposed a method to effectively analyze the authentication scheme and check to see if the scheme is logically workable. Similarly, in this section, we shall use the BAN logic to prove the logical correctness of our authentication procedure.

This BAN logic check mainly consists of 4 parts:

1. Setting verification goals.
2. Converting generic type to idealized form.
3. Making assumptions.
4. Analyzing the proposed scheme.

A. Verification Goals

The verification goals are as follows:

{G1} $U_i$believes$U_i \overset{SK}{\longleftrightarrow} AS$

{G2} $U_i$believes *AS* believes $U_i \overset{SK}{\longleftrightarrow} AS$

{G3} *AS* believes $U_i \overset{SK}{\longleftrightarrow} AS$

{G4} *AS* believes U_i believes $U_i \overset{SK}{\longleftrightarrow} AS$

B. Idealized Form

The idealized type of our scheme is as follows:

M1:$U_i \rightarrow AS : (< \alpha >_{C_i}, T_1)$

M2:$AS \rightarrow U_i : (<U_i \overset{SK}{\longleftrightarrow} AS, \beta >_{C_i}, T_1 + 1)$

M3:$U_i \rightarrow AS : (<U_i \overset{SK}{\longleftrightarrow} AS> SK, T_1 + 2)$

C. Assumptions

The assumptions are as follows:

{A1} $U_i$believes fresh T_1

{A2} AS believes fresh $T_1 + 1$

{A3} AS believes U_i controls α

{A4} $U_i$believes AS controls β

{A5} AS believes $U_i \overset{C_i}{\longleftrightarrow} AS$

{A6} $U_i$believes $U_i \overset{C_i}{\longleftrightarrow} AS$

{A7} $U_i$believes AS controls $U_i \overset{SK}{\longleftrightarrow} AS$

{A8} AS believes U_i controls $U_i \overset{SK}{\longleftrightarrow} AS$

{A9} AS believes $U_i \overset{SK}{\longleftrightarrow} AS$

{A10} $U_i$believes $U_i \overset{SK}{\longleftrightarrow} AS$

D. Analysis of Proposed Scheme by BAN Logic

Based on the BAN logic rules and above-mentioned suppositions, the primary steps of proof are as per the following:
By M1 and the rule of seeing, we can derive the following statement

AS sees $(< \alpha > C_i, T_1)$ →{S1}

By {S1}, {A5} and rule of message meaning,

AS believes U_i said (α, T_1) →{S2}

By {S2}, {A1}, rule of nonce verification and rule of freshness,

AS believes U_i believes α →{S3}

By {S3}, {A3} and rule of jurisdiction,

AS believes α →{S4}

By M2 and rule of seeing, we can derive the following statement

$U_i$sees $(< U_i \overset{SK}{\longleftrightarrow} AS, \beta > C_i, T_1 + 1)$ →{S5}

By {S5}, {A6} and rule of message meaning,

$U_i$believes AS said $(U_i \overset{SK}{\longleftrightarrow} AS, \beta, T_1 + 1)$ →{S6}

By {S6}, {A2}, rule of nonce verification and rule of freshness,

U_i believes AS believes $(U_i \overset{SK}{\longleftrightarrow} AS, \beta, T_1 + 1)$ →{S7}

By {S7} and breaking the conjunction,

$U_i$believes AS believes $U_i \overset{SK}{\longleftrightarrow} AS$ →{S8}

By {S8}, {A7} and rule of jurisdiction,

$U_i$believes $U_i \overset{SK}{\longleftrightarrow} AS$ →{S9}

By M3 and rule of seeing, we can derive the following statement

AS sees $(< U_i \overset{SK}{\longleftrightarrow} AS > SK, T_1 + 2)$ →{S10}

By {S10}, {A9} and rule of message meaning,

AS believes $U_i$said $(U_i \overset{SK}{\longleftrightarrow} AS, T_1 + 2)$ →{S11}

By {S11}, {A1}, rule of nonce verification and rule of freshness,

AS believes $U_i$believes $(U_i \overset{SK}{\longleftrightarrow} AS, T_1 + 2)$ →{S12}

By {S12} and breaking the conjunction,

AS believes $U_i$believes $U_i \overset{SK}{\longleftrightarrow} AS$ →{S13}

By {S13}, {A8} and Jurisdiction rule,

AS believes $U_i \overset{SK}{\longleftrightarrow} AS$ →{S14}

The statements {S9}, {S8}, {S14} and {S13} put together satisfy the verification goals {G1}, {G2}, {G3} and {G4} of the projected scheme. In view of these announcements, the proposed scheme is capable of setting up a safe session key among U_i and AS. Hence both U_i and AS are able to authenticate each other using this scheme.

VII. ANALYSIS OF THE PROPOSED SCHEME

In this section, we shall discuss the security of the projected scheme, demonstrating that our new scheme not only inherits the strengths of Islam's scheme but can solve the security problems found in Islam's [19] scheme. In addition, our scheme will also be compared with several similar schemes in terms of computation cost.

A. Security Analysis

These theorems have been established to validate the security features of our scheme:

Theorem 1: Our scheme can withstand the off-line/on-line password guessing attack and stolen/lost *SC* attack.

Proof: Many researchers have claimed that the information put away in the *SC* can be separated in many ways for example, power consumption analysis etc. [1, 4, 10, 11, 12, 15, 22, 24]. Assume that an attacker *E* robs $U_i$of his/her SC and collects the information $\{h(.), n, A_i, B_i, R_i, S_i\}$, where $B_i = C_i \oplus h(PW_i||b_i||sid_i)$, $A_i = h(b_i||sid_i||PW_i||C_i)$, $R_i = b_i \oplus h(PW_i||sid_i)$, and $S_i = T_i \oplus h(sid_i||b_i||PW_i)$. The attacker *E*still cannot derive U_i'spassword from the above equations since he/she does not know $sid_i, PW_i, C_i$and b_i. If the attacker knew all necessary values except PW_i, then there might still be a slight chance of guessing it right. However, guessing more than one value (i.e $.(PW_i||sid_i)$, or $(PW_i||b_i||sid_i)$, or $(b_i||sid_i||PW_i||C_i)$)at the same time is not possible. Therefore, we can claim that our scheme is protected against the off-line password guessing attack and stolen/lost *SC* attack.

In an online password guessing attack, the attacker tries to login to the server by entering one term after another from a dictionary in an attempt to match the user's login sid and PW.This kind of attack is basically not workable because the task of guessing a single value within polynomial time (i.e., Δt) is generally considered impossible, let alone when there are more than one variable to deal with at the same time (e.g., $(PW_i||sid_i)$, or $(PW_i||b_i||sid_i)$, or $(b_i||sid_i||PW_i||C_i)$). The attacker is entitled to only a maximum of three trials, and if all three have failed, the SC gets locked up. Therefore, there is no way the on-line *PW* guessing attack can take effect on our scheme.

Theorem 2: The projected scheme can withstand the recognized session-specific temporary information attack.

Proof: Let us consider if the session-specific random numbers (α, β) chosen by U_i and AS should be compromised by an attacker. In that situation, the attacker still has not way to derive the session key $SK = (D_i)^{\beta sid_i}(mod\ n^2)$ (or) $V_i^{(\alpha.b_i)}(mod\ n^2)$ (or) $(C_i)^{\alpha b_i \beta sid_i}(mod\ n^2)$. The attacker mayhave a chance to obtain $D_i$or $V_i$over a public communication channel, but to frame *SK* he/she needs to havesid_i, or b_i, or C_i,

which he/she does not, along with α, β. In this manner, we can say that our scheme can resist the known session-specific temporary information attack.

Theorem 3: Our scheme gives the security feature of session key perfect forward secrecy.

Proof: To claim that our scheme has this feature, we have to prove that no session keys would be revealed even if the server's private key x should be known to some attacker E. In our scheme, U_i and AS create $SK = (C_i)^{\alpha b_i \beta s i d_i} (mod\ n^2)$, where $C_i = h(\mathrm{s}id_i||x||\mathrm{s}s id_i)$. The attacker E cannot derive SK from the eavesdropped message $\{D_i, V_i\}$ even if x is at hand, for E does not have $\mathrm{s}id_i$ and b_i.

Theorem 4: Our scheme is protected against the known key attack using session-particular random numbers.

Proof: Generally speaking, the pair of user and server will share a common SK for each session. To offer proper protection against the known key attack, we have to make sure that the current SK can never be derived from earlier session keys. In other words, with one SK somehow leaked out, we still have to guarantee the safety of the future and/or previous session keys. In our scheme, if the $SK = (C_i)^{\alpha b_i \beta s i d_i} (mod\ n^2)$ of a present session should leak out, the attacker can still not use this information to reveal other SK's because the session-specific random numbers α, β are different for different sessions, and also $\mathrm{s}id_i$ and b_i are unknown to E.

Theorem 5: Our scheme can withstand the forgery/modification attack and the user/server masquerade attack.

Proof: Assume that an attacker has had some messages traveling between user and server (i.e., login & authentication information) intercepted and now has $\{D\mathrm{s}id_i, D_i, M_i, T_i, T_1\}$ and $\{\mathrm{s}Ds id_i;\ V_i;\ M_s;\ T_2;\ T_i^{new}\}$, where $M_i = h(\mathrm{s}id_i||C_i||D_i||\ T_i||\ T_1)$ and $M_s = h(\mathrm{s}id_i||\ C_i^*||V_i||SK||T_2)$. The attacker E cannot mess with M_i and M_s because he/she does not have C_i. Therefore, we claim that our scheme is protected against the forgery/modification attack.

To impersonate a user, the attacker E must frame a login request message $\{D\mathrm{s}id_i, D_i, M_i, T_i, T_1\}$ correctly. The attacker needs to have $\mathrm{s}id_i$ and $\mathrm{s}s id_i$ to be able to frame $D\mathrm{s}id_i = \mathrm{s}id_i \oplus h(\mathrm{s}s id_i\ ||\ T_1||\ T_i)$. However, these two values are unknown to E. This demonstrates our scheme is protected against a user masquerade attack.

To imitate a server, the attacker must frame a login answer message $\{\mathrm{s}Ds id_i;\ V_i;\ M_s;\ T_2;\ T_i^{new}\}$ correctly. The attacker needs to have $\mathrm{s}id_i$ and C_i to frame $\mathrm{s}Ds id_i$, V_i and M_s. However, these two values are unknown to the attacker. This proves that our scheme can oppose a server masquerade attack.

Theorem 6: Our scheme provides user anonymity.

Proof: User anonymity means the identity $\mathrm{s}id_i$ of a user U_i must be under proper protection so that no attacker has access to it and can relate it to possible passwords. In our proposed scheme, the user's $\mathrm{s}id$ is communicated over a public communication channel, so the attacker E has no way to obtain $\mathrm{s}id_i$. That means our scheme satisfies the user anonymity requirement.

Theorem 7: Our scheme can resist the replay attack.

Proof: This kind of attack happens when an attacker tries to login to the server by sending caught earlier messages among legal user and server. In our scheme, replaying messages of one session to another session will not work because the user's *SC* and the server use the current time stamps T_1 and T_2 in each new session, which means the values of $M_i, D\mathrm{s}id_i$, and M_s are dynamic. The value of T_i is also dynamic in every session and will be updated both in the user's *SC* and the server's database. Hence our proposed scheme is protected against the message replay attack.

B. Computation Cost Analysis

In this analysis, there are two notations of time complexity we put to use, which are:

T_e: Time taken to execute one exponent operation

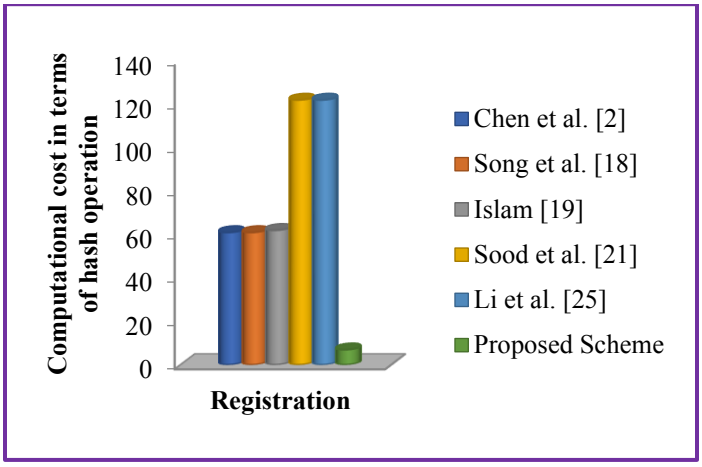
T_h: Time taken to execute one hash operation.

Table III shows how our new scheme compares with several other similar schemes [2, 5, 17, 19, 18, 21, 25] regarding computation cost. As shown in the table, our scheme desires to do a total of 4 exponential operations and 37 hash functions. Please notice that hash operations are always much lower in computation cost than exponential operations with the cost for one exponent operation approximately equal to that for 60 hash operations. Even though the projected scheme involves quite a large number of hash operations, the numbers of exponential operations needed have been cut down to only one for login and three for authentication, making the proposed scheme the most efficient of them all [2, 5, 17, 19, 18, 21, 25]. The relations between T_h and T_e ($T_h < T_e, T_e \approx 600\ T_h$) with respect to $T_h = 0.503\ (ms)$ have been established in [30], [28], [34], [35].

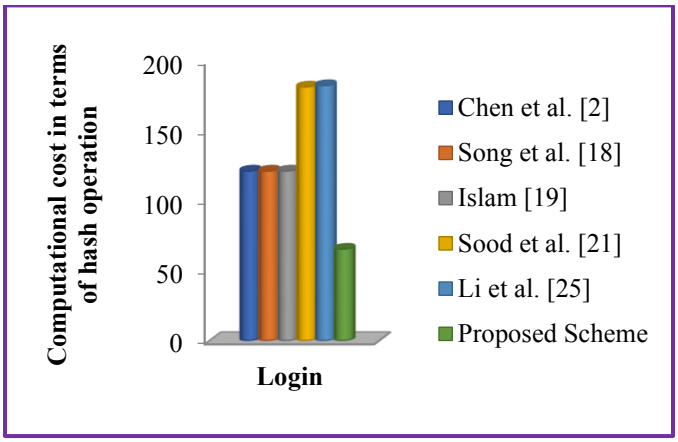
Table II. Functionality examination of the projected scheme with other similar schemes

Security features	**[2]**	**[18]**	**[19]**	**[21]**	**[25]**	**Proposed Scheme**
Stolen SC attack	√	No (×)	×	Yes (√)	×	√
Insider attack	×	×	×	×	×	√
Perfect forward secrecy	×	×	√	×	√	√
Detection of wrong password	×	×	√	×	√	√
User impersonation attack	×	×	×	×	√	√
Cancellation of lost smartcard	×	×	√	×	×	√
Mutual authentication	√	√	×	×	√	√
Session specific temporary information attack	×	×	×	×	×	√

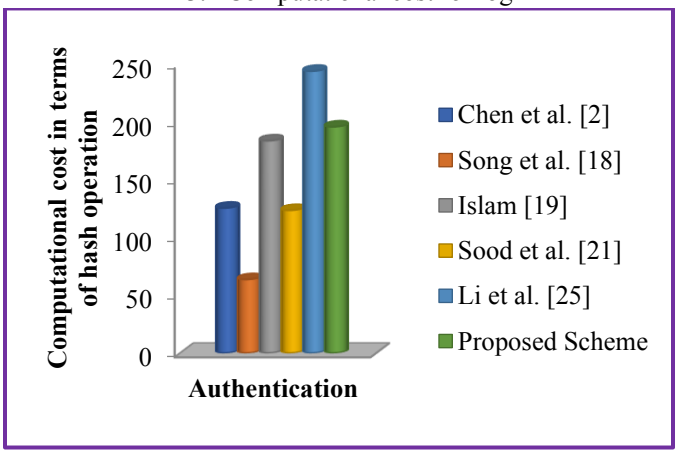
Figure 3 show the efficiency of the proposed scheme over the existing schemes in the literature. The visual representations shown in Figure 3.1, Figure 3.2, and Figure 3.4 show that the proposed scheme require less computational cost as compared to the existing schemes in terms of registration, login, and password change steps. It has also seen from Figure 3.2 that the proposed scheme is effective as compared to some of the schemes in terms of computational cost require for authentication step.



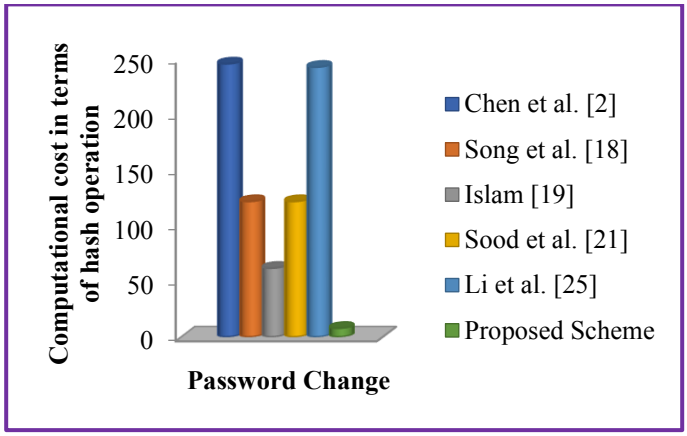
3.1 Computational cost for registration



3.2 Computational cost for login



3.3 Computational cost for Authentication



3.4 Computational cost for Password change

Fig. 3 Quantitative analysis based on the computational cost for each operation

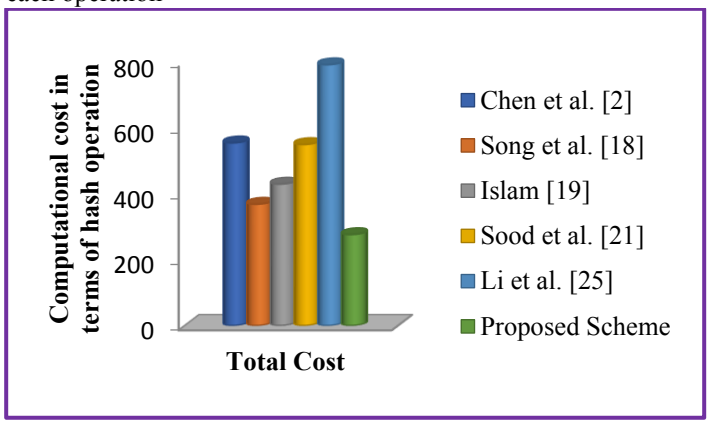


Fig. 4. Quantitative analysis based on the total computational cost

Figure 4 show that, the proposed scheme is efficient than the existing schemes with respect to the total computational cost.

Table III. Computation cost examination of the projected scheme with other similar schemes

Schemes/ Phases	Registration	Login	Authentication	PW change
[2]	$1T_h + 1T_e = 61h$	$2T_h + 2T_e = 122h$	$6T_h + 2T_e = 126h$	$6T_h + 4T_e = 246h$
[18]	$1T_h + 1T_e = 61h$	$2T_h + 2T_e = 122h$	$4T_h + 1T_e = 64h$	$2T_h + 2T_e = 122h$
[19]	$2T_h + 1T_e = 62h$	$2T_h + 2T_e = 122h$	$4T_h + 3T_e = 184h$	$2T_h + 1T_e = 62h$
[21]	$2T_h + 2T_e = 122h$	$2T_h + 3T_e = 182h$	$4T_h + 2T_e = 124h$	$2T_h + 2T_e = 122h$
[25]	$2T_h + 2T_e = 122h$	$3T_h + 3T_e = 183h$	$4T_h + 4T_e = 244h$	$3T_h + 4T_e = 243h$
Proposed Scheme	$7T_h = 7h$	$6T_h + 1T_e = 66h$	$16T_h + 3T_e = 196h$	$8T_h = 8h$

Note: One exponent operation is approximately equal to 60 hash operations

VIII. CONCLUSION

This paper identified the security weaknesses of Islam's scheme including vulnerability to the offline password guessing attack, stolen *SC* attack, user impersonation attack, and known session-specific temporary information attack, as well as failure to preserve user anonymity. To solve those problems, we have introduced an enhanced scheme based on sub-tree and partial discrete logarithm problem for fuzzy user under cloud computing that is more efficient, lower in computation cost, and, most importantly, brings smartcard-based password authentication to a higher level of security.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers of International Journal of Circuits, Systems and Signal Processing for their careful and helpful comments.

References

[1] A. K. Das, V. Odelu, and A. Goswami, "A Robust and Effective Smart-Card-Based Remote User Authentication Mechanism Using Hash Function", The Scientific World Journal, vol. 2014, (2014), Article ID 719470, 16 pages, doi.org/10.1155/2014/719470.

[2] B. L. Chen, W. C. Kuo, L. C. Wu, "Robust smartcard-based remote user password authentication scheme". International Journal of Communication Systems, vol. 27, (2014), pp. 377–389.

[3] C. Chen, D. J. He, S. M. Chan, J. J. Bu, Y. Gao, R. Fan, "Lightweight and provably secure user authentication with anonymity for the global mobility network". International Journal of Communication Systems, vol. 24(3), (2011), pp. 347–362.

[4] C-C Chang, T-F Cheng and W-Y Hsueh, "A robust and efficient dynamic identity-based multi-server authentication scheme using smart cards", International Journal of Communication Systems, vol. 29(2), (2016), pp. 290–306.

[5] C. G. Ma, D. Wang, S-D Zhao, "Security flaws in two improved remote user authentication schemes using smart cards", International Journal of Communication Systems vol. 27(10), (2014), pp. 2215–2227.

[6] C. M. Swanson, "Security in key agreement: two-party certificateless schemes", Master's thesis, University of Waterloo, Canada, 2008.

[7] H. B. Tang, X. S. Liu, "Cryptanalysis of a dynamic ID-based remote user authentication with key agreement scheme", International Journal of Communication Systems, vol. 25(12), (2012), pp. 1639–1644.

[8] J. Xu, W. T. Zhu, D. G. Feng, "An improved smartcard-based password authentication scheme with provable security", Computer Standards and Interfaces, vol. 31(4), (2009), pp. 723–728.

[9] K. H. Yeh, N. W. Lo, Y. J. Li, "Cryptanalysis of Hsiang–Shih's authentication scheme for multi-server architecture", International Journal of Communication Systems, vol. 24(7): (2011), pp. 829–836.

[10] M. Burrows, M. Abadi and R. Needham, "A Logic of Authentication", ACM Transactions on Computer Systems, vol. 8(1), 1990, pp. 18-36.

[11] M. S. Farash, "An improved password-based authentication scheme for session initiation protocol using smart cards without verification table", International Journal of Communication Systems, (2014), DOI: 10.1002/dac.2879

[12] M. Joye, F. Olivier, "Side-Channel Analysis, Encyclopedia of Cryptography and Security", Kluwer Academic Publishers: Springer USA, (2005), pp. 571–576.

[13] M. Hou, Q. Xu, G. Shanqing, H. Jiang, "Cryptanalysis of identity-based authenticated key agreement protocols from parings", Journal of Networks, vol. 5(7), (2010), pp. 826–855.

[14] M K Khan, S K Kim, K. Alghathbr, "Cryptanalysis and security enhancement of a 'more efficient & secure dynamic ID-based remote user authentication scheme", Computer Communications, vol. 34, (2011), pp. 305–309.

[15] P. Kocher, J. Jaffe, B. Jun, "Differential power analysis", Proceedings of Advances in Cryptology (Crypto'99), LNCS, Springer Berlin Heidelberg, (1999), pp. 388–397.

[16] R. Canetti, H. Krawczyk, "Analysis of key exchange protocols and their use for building secure channels", Proceedings of Advances in Cryptology (Eurocrypt'01), London, UK, (2001), pp. 453–474.

[17] R. Song, "Advanced smartcard-based password authentication protocol", Computer Standards and Interfaces, vol. 32(5), (2010), pp. 321–325.

[18] R. Song, L. Korba, G. Yee, "Analysis of smart card-based remote user authentication schemes", Proceedings of the 2007 International Conference on Security and Management, Las Vegas, USA, (2007), pp. 323–329.

[19] S. K. Hafizul Islam. "Design and analysis of an improved smartcard–based remote user password authentication scheme", International Journal of Communication Systems, vol. 29, no. 11, (2016), pp. 1708-1719.

[20] S. Kumari, M. K. Khan, "Cryptanalysis and improvement of 'a robust smart-card-based remote user password authentication scheme", International Journal of Communication Systems, vol. 27(12), (2014), pp. 3939–3955.

[21] S. K. Sood, A. K. Sarje, K. Singh, "An improvement of Xu et al.'s authentication scheme using smartcards", Proceedings of the Third Annual ACM Bangalore Conference, Bangalore, Karnataka, India, (2010), pp. 17–25.

[22] T. Jain and S. P. Singh, "An Efficient and Secure Multi-Server Smart Card based Authentication Scheme", International Journal of Computer Applications, vol. 93(13), (2014), pp. 1-7.

[23] T. Mandt, C. Tan, "Certificateless authenticated two-party key agreement protocols", Proceedings of the ASIAN, vol. 4435, Springer-Verlag, Springer Berlin Heidelberg, (2008), pp. 37–44.

[24] T S Messerges, E. A. Dabbish, R. H. Sloan, "Examining smart card security under the threat of power analysis attacks", IEEE Transactions on Computers, vol. 51(5), (2002), pp. 541–552.

[25] X. Li, J. Niu J, M. K. Khan, J. Liao, "An enhanced smartcard based remote user password authentication scheme", Journal of Network and Computer Applications, vol. 36, (2013), pp.1365–1371.

[26] W. Liu, J. Liu, Q. Wu, B. Qin, D. Naccache, H. Ferradi, "Efficient subtree-based encryption for fuzzy-entity data sharing", Soft Computing, vol. 22 (23) (2018), pp.7961–7976.

[27] C. Meshram, C.C. Lee, S. G. Meshram, M. K. Khan, "An Identity-based encryption technique using subtree for fuzzy user data sharing under cloud computing environment", Soft Computing, vol. 23(24), (2019), pp. 113127–13138.

[28] C. Meshram, M.S. Obaidat, S.G. Meshram, "Chebyshev Chaotic Maps based ID-based Cryptographic Model using Subtree and Fuzzy-entity Data Sharing for Public Key Cryptography", Security and Privacy, **1(1) e12**, (2018), pp. 1-9.

[29] P. Paillier, "Public key cryptosystem based on discrete logarithm residues," In Eurocrypt' 99, Lecture Notes in Computer Science, vol. 1592, (1999), pp. 223-238.

[30] C. Meshram, M. S. Obaidat, C-C. Lee, S. G. Meshram, "An Efficient Key Authentication Procedure for IND-CCA2 Secure Paillier-based Cryptosystem", Soft Computing, vol. 24 (9), (2020), pp. 6531–6537.

[31] C. Meshram, P. L. Powar, M. S. Obaidat and Cheng-Chi Lee, "An IBE Technique using Partial Discrete Logarithm", Procedia Computer Science, vol. 93, (2016), pp. 735-741.

[32] C. Meshram, P. L. Powar and M. S. Obaidat, "An UF-IBSS-CMA Protected Online/Offline Identity-based Short Signature Technique using PDL", Procedia Computer Science, vol. 93, (2016), pp. 847-853.

[33] Z. Cheng, M. Nistazakis, R. Comley, L. Vasiu. "On the indistinguishability-based security model of key agreement protocols-simple cases", Cryptology ePrint Archieve, Report 2005/129, 2005.

[34] C. Meshram, Chun-Ta Li, S. G. Meshram, "An Efficient Online/Offline ID-based Short Signature Procedure using Extended Chaotic Maps", Soft Computing, vol. 23(3), (2019), pp.747-753.

[35] C. Meshram, Cheng-Chi Lee, S. G. Meshram, Chun-Ta Li, "An Efficient ID-based Cryptographic Transformation Model for Extended Chaotic-Map-Based Cryptosystem", Soft Computing, vol. 23 (16), (2019), pp. 6937–6946.

[36] N. Scheidt, M. Adda, "Framework of Confidence Values during Digital Forensic Investigation Processes", WSEAS Transactions on Systems and Control, vol. 15, (2020), pp. 228-234.

[37] I. Ganchev, Z. Ji, M. O'Droma, "Designing a Cloud Tier for the IoT Platform EMULSION", WSEAS Transactions on Systems and Control, vol. 14 (2019) pp. 375-383.

First A. Author (M'76–SM'81–F'87) and the other authors may include biographies at the end of regular papers. Biographies are often not included in conference-related papers. This author became a Member (M) of NAUN in 1976, a Senior Member (SM) in 1981, and a Fellow (F) in 1987. The first paragraph may contain a place and/or date of birth (list place, then date). Next, the author's educational background is listed. The degrees should be listed with type of degree in what field, which institution, city, state or country, and year degree was earned. The author's major field of study should be lower-cased.

The second paragraph uses the pronoun of the person (he or she) and not the author's last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (city, state: publisher name, year) similar to a reference. Current and previous research interests ends the paragraph.

The third paragraph begins with the author's title and last name (e.g., Dr. Smith, Prof. Jones, Mr. Kajor, Ms. Hunter). List any memberships in professional societies other than the NAUN. Finally, list any awards and work for NAUN committees and publications. If a photograph is provided, the biography will be indented around it. The photograph is placed at the top left of the biography. Personal hobbies will be deleted from the biography.

Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

Author Contributions: Please, indicate the role and the contribution of each author:
Example
Chen Lee carried out the simulation and the optimization.
Kemal Mehmet has implemented the Algorithm 3.2 and 15.1 in Java
George Luton has organized and executed the experiments of Section 4.
Michael Walton was responsible for the Simulation and Statistics.
In general, please, follow
http://naun.org/main/format/contributor-role.pdf

Sources of funding for research presented in a scientific article or scientific article itself

Report potential sources of funding if there is any

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)