An Anonymous and Efficient Multiserver Authenticated Key Agreement With Offline Registration Centre

Azeem Irshad, Shehzad Ashraf Chaudhry, Muhammad Sher, Bander A. Alzahrani, Saru Kumari, Xiong Li, and Fan Wu

***Abstract*—Multiserver authentication (MSA) architecture permits its registered users to get the services of various service providers without performing separate registrations with each server. Since, a single registration from registration authority, a trusted third party, is sufficient to get mutually authenticated with service providers, onwards. In this paper, we have emphasized only on the MSA-based schemes that do not require online registration authority for mutual authentication. While, considering those MSA schemes, to date, there is no secure and efficient MSA protocol in our observation that is free of all four limitations at the same time, such as 1) free of the hassle of storage of server-based parameters in users' smart card by registration authority, 2) free of the transmission of user-based identities to all servers in a network, 3) free of a single secret key distribution to all servers as assumed in a trusted system, and 4) free from employing costly bilinear pairing operations. In view of those shortcomings, we present an anonymous MSA protocol that nearly counters all those problems using elliptic curve cryptography and escaping costly pairing operations. Our scheme also demonstrates the scheme's robustness using formal security analysis employing random oracle model and automated tool analysis.**

***Index Terms*—Attacks, cryptanalysis, multiserver authentication (MSA), registration centre, security, trusted third party.**

I. Introduction

AUTHENTICATION is deemed to be a critical component for any secure transaction. The authentication schemes have evolved from password-based schemes to smart card and onward to biometric factor-based schemes. Those password-based schemes [1], [3] require servers to maintain password-verifier tables. While, two-factor-based smart card authentication schemes were introduced [2], [10], [12], [30], to evade verification tables for its demerits. However, stolen smart cards induced many offline-password guessing or spoofing attacks. Later, three-factor biometric authentication schemes [5], [9], [11] were adopted to counter the weaknesses of the former type of schemes.

Multiserver authentication (MSA) relinquishes the users' requirement of performing registration with each server in a network. Following this, the users depend on a single registration of a registration centre for getting the services of various servers. MSA alleviates the burden of servers of performing separate registration of users. The MSA architecture comprises three interacting members, i.e., registration centre RC, user Ui, and server Sj. The RC, being an online registration authority registers all users and servers using secure channels in the initialization phase. The schemes in MSA can be classified into two types: i.e., 1) schemes involving Online Registration Centre in mutual authentication phase (MRC-On) and 2) schemes not employing Online Registration Centre (MRC-Off) during login and authentication, as shown in Fig. 1. The MRC-Off schemes were established in order to alleviate the numerous visits to online RC during mutual authentication phase. Since, an online RC may become a bottleneck, if it is burdened with lots of authentication requests, and ultimately a user suffers in terms of delay.

In this study, we specifically focus on the drawbacks in MRC-Off schemes and suggest an authentication scheme countering the flaws in existing MRC-Off studies. Currently, there is no proficient MRC-Off protocol in our analysis that is free either from verifiers' database maintenance (maintained at central authority level during registration phase), or storing the servers' identifiers or public keys in users' smart cards, or submitting the users' identifiers to each server after user registration, or free from sharing a single key with all servers in a trusted system, simultaneously. Since the management of any verifiers' repository may be tough, or even its use may lead to Denial-of-service attack. At the same time, storing identifiers either in smart cards or transmitting to servers affects the scalability of the system. While, sharing a single secret in a trusted system may lead to server masquerading attacks. On the other side, the bilinear

Manuscript received June 28, 2017; revised April 2, 2018; accepted May 13, 2018. Date of publication June 20, 2018; date of current version February 22, 2019. This work was supported in part by the Scientific Research Fund of Hunan Provincial Education Department under Grant 16B089 and in part by the Hunan Provincial Natural Science Foundation of China under Grant 2018JJ3191. *(Corresponding author: Azeem Irshad.)*

A. Irshad, S. A. Chaudhry, and M. Sher are with the Department of Computer Science and Software Engineering, International Islamic University, Islamabad 46000, Pakistan (e-mail: irshadazeem2@gmail.com; shahzad@iiu.edu.pk; m.sher@iiu.edu.pk).

B. A. Alzahrani is with the Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia (e-mail: baalzahrani@kau.edu.sa).

S. Kumari is with the Department of Mathematics, Chaudhary Charan Singh University, Meerut 250004, India (e-mail: saryusiirohi@gmail.com).

X. Li is with the Hunan University of Science and Technology, China (e-mail: lixiongzhq@163.com).

F. Wu is with the Xiamen Institute of Technology, Xiamen, China (e-mail: conjurer1981@gmail.com).

Digital Object Identifier 10.1109/JSYST.2018.2838450

1937-9234 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: Abu Dhabi University. Downloaded on March 13,2025 at 11:15:12 UTC from IEEE Xplore. Restrictions apply.

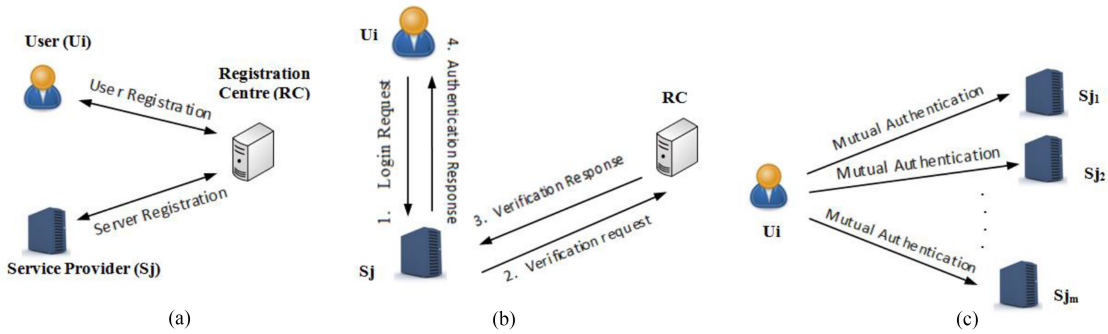

Fig. 1. Registration and mutual authentication in MSA-architecture. (a) Registration phase. (b) Mutual authentication (MRC-On). (c) Mutual authentication (MRC-Off).

pairing-based MSA techniques employ costly operations to provide a secure MRC-Off scheme. In this scenario, there is still a need of more robust MRC-Off scheme countering the identified limitations.

A. Related Work

The traditional single-server authentication techniques [6]–[8] are unable to meet up to the demands of an urbane user, which emphasizes the need of another multiserver-based paradigm. The MSA schemes can be traced back to Lee *et al.* and Li *et al.*, in 2000 [2], [3], and then various protocols [4], [13]–[25] have been presented onward that suffered many limitations. Although, we can witness several MRC-On-based protocols [22], [40]–[44], nonetheless these protocols bear high latency due to more communication rounds, which can be attributed to involvement of online RC during login and authentication phase. In this study, we shall critically examine into the literature of MRC-Off schemes that would help us analyzing and evaluating the comparative studies with the proposed protocol.

B. Limitations of MRC-Off-Based Schemes

The MRC-Off-based schemes can be classified into four types on the basis of functionality:

1) $SSSS$: The Schemes Sharing a Single Secret with all servers that may lead to malicious server spoofing attack.
2) $SSSC$: The Schemes in which the Servers' public keys or verifiers are stored in the Smart Cards of the users.
3) $SUSP$: The Schemes in which that public keys or verifiers of Users should be transmitted to all Service Providers in a network.
4) $SCBP$: The Schemes relying on Costly Bilinear Pairing operations.

For the sake of understanding, we describe drawbacks of MRC-Off-based protocols by using the same notations as assumed in original schemes. The scheme [28] ($SSSS$) shares secret "y" with all servers and users that renders the scheme vulnerable for user and server impersonation attacks. In [29] and [31] ($SSSS$), the trusted authority shares $h(x||y)$ and $h(y)$ with all service providers leading to server masquerading attack. In [32] ($SSSS$), the trusted authority shares keys $k_1 = h(K||b)$ and $k_2 = h(SIDj||PS)$ with all servers. In [30], [33], [34] ($SSSS$), all servers share PSK (Preshared Key) which may lead to server impersonation attack. In [35]–[37], [49] schemes ($SSSC$), the servers' specific parameters are stored in smart cards during registration phase. However, adding new service provider into the system involves massive update in users' smart cards, which makes the system unscalable for a large network. The scheme [38] ($SUSP$) assumes that the public keys and users' verifiers are transmitted to all servers in the network so that servers could maintain verifiers' database of legitimate users and verify those, whenever needed. It again affects the system's scalability for adding users into the system and updating servers' repository thereafter. In [46]–[48] schemes ($SCBP$), the costly bilinear pairing operations augment the protocol's computational cost. In view of these concerns in MRC-Off-based protocols, we propose a robust MRC-Off-based scheme that does not maintains any verification table, neither stores any server or user-based parameter in smart cards or server's repository, respectively, nor shares a single secret with servers to identity users, yet the participants mutually authenticate one another in an efficient manner.

C. Layout of the Manuscript

As per the layout of the proposed scheme, Section II depicts preliminaries for basic cryptography. Section III demonstrates our proposed model. Section IV presents security discussion, automated tool analysis, and formal security analysis using random oracle model. Section V exhibits performance evaluation of proposed and compared schemes. The last section gives the conclusion.

II. Preliminaries

The preliminary section describes the properties of hash function, elliptic curve cryptography, and bio-hashing as used in the proposed contribution.

A. One-Way Hash Function

Given, η be the whole set of binary digit sequences of indefinite length, and η' be the whole set of binary sequences of fixed length ι, a secure one way hash-based function h_{fn}: $(\eta \rightarrow \eta')$ outlines the under-mentioned properties:

Authorized licensed use limited to: Abu Dhabi University. Downloaded on March 13,2025 at 11:15:12 UTC from IEEE Xplore. Restrictions apply.

1. The hash-based function h_{fn} takes a character string of indefinite length and outputs a digest of fixed length;
2. Given h_{fn}: ($\eta \rightarrow \eta'$), it is not likely to calculate $h_{fn}^{-1}(\eta') = \eta$ in polynomial time;
3. Given η, it is hard to craft η'', such that $\eta'' \neq \eta$, nonetheless $h_f\ (\eta'') = h_{fn}\ (\eta)$;
4. It is hard, computationally, to get any pair η, η'' such that $\eta'' \neq \eta$, and $h_{fn}\ (\eta'') = h_f\ (\eta)$.

Definition 1: $Adv_{\mathcal{A}}^{h_{fn}}(\tau_1)$ be the advantage of a polynomial time-bounded adversary $\mathcal{A}$ in τ_1 time for $\Xi, \partial\ \epsilon\, \mathcal{U}$ such that $\mathrm{h_f}(\Xi) = \mathrm{h_f}\ (\partial)$, then the probability of advantage is $Adv_{\mathcal{A}}^{h_{fn}}(\tau_1)$ for $\mathcal{A}$ in random picks of τ_1 time. Thus, the function $h_{fn}()$ acts collision-resistant, only if the advantage $Adv_{\mathcal{A}}^{h_{fn}}(\tau_1) \leq \mu_1$ for some negligible function (μ_1), i.e.,

$$Adv_{\mathcal{A}}^{h_{fn}}\ (\tau_1) = \Pr[\,(\Xi, \partial) \Leftarrow_R \mathcal{U} \times \mathcal{U} |\ (\Xi \neq \partial) \wedge h_{fn}\ (\Xi) = h_{fn}\ (\partial)] \quad (1)$$

B. Elliptic Curve Cryptography

We define a prime finite field as F_p and p as high entropy prime integer. The solution set (δ, σ) $\epsilon\, F_p \times F_p$ for equation $\sigma^2 = (\delta^3 + \Upsilon\delta + \Omega) \bmod p$ with "O" serving as *infinity point* defines a set E_c, known as nonsingular elliptic curve, where Υ, $\Omega\ \epsilon\ F_p$, and $(4\ \Upsilon^3 + 27\ \Omega^2) \neq 0$. The strength of our scheme is rooted in the hardness of the two computational problems:

Definition 2: Given λ and P, associated as $\lambda = \Xi$ P, where λ and P ϵ $\mathrm{E_C}$ (δ, σ). It is hard to recover $\Xi\ \epsilon\ \mathrm{Z^*_p}$, If $\mathcal{A}$'s advantage to recover Ξ from P and λ in random time τ_2 refers to $Adv_{\mathcal{A}}^{\mathrm{ECDLP}}(\tau_2)$, in that case we have $Adv_{\mathcal{A}}^{\mathrm{ECDLP}}(\tau_2)$ as the probability of $\mathcal{A}$'s advantage as calculated by attacker in τ_2. We refer ECDLP as hard, provided that $Adv_{\mathcal{A}}^{\mathrm{ECDLP}}(\tau_2) \leq \mu_2$, for any negligible function (μ_2). The gain of $\mathcal{A}'s$ advantage $Adv_{\mathcal{A}}^{\mathrm{ECDLP}}(\tau_2) \leq \mu_2$ can be drawn as given in the following:

$$Adv_{\mathcal{A}}^{\mathrm{ECDLP}}\ (\tau_2) = \Pr[\Xi \epsilon Z_p^* | \lambda = \omega P]. \quad (2)$$

Definition 3: By the definition of an elliptic curve based computational Diffie–Hellman problem (ECDHP), it is hard to devise $[\Xi.\partial]P\epsilon\, E_C(\delta, \sigma)$ out of parameters $\lambda, \mho, P$ given $\lambda = [\Xi]$ P, $\mho = [\partial]$ P, while $P, \lambda, \mho \epsilon E_C(\delta, \sigma)$ and $\Xi, \partial \epsilon Z_p^*$. The advantage prospects of $\mathcal{A}$ to compute $[\Xi.\partial]P$ in time duration τ_3 amounts to $Adv_{\mathcal{A}}^{\mathrm{CDHP}}(\tau_3)$. The CDHP problem is hard if $Adv_{\mathcal{A}}^{\mathrm{CDHP}}(\tau_3) \leq \mu_3$, for some negligible function (μ_3). The equation for advantage $Adv_{\mathcal{A}}^{\mathrm{CDHP}}(\tau_3) \leq \mu_3$ is shown as under

$$Adv_{\mathcal{A}}^{\mathrm{CDHP}}\ (\tau_3) = \Pr[\,[\Xi.\partial]\, P\epsilon E_C\ (\delta, \sigma)\ |\lambda = [\Xi]\ P \wedge \mho = [\partial]\, P]. \quad (3)$$

C. Bio-Hashing

The biometric features are inalienable in personal identity-based associations. One of the biometric authentication system, termed as bio-hashing function by Lumini *et al.* [27] transforms real-value data into binary strings and assigns a single bit to generated random projections. Thereby, the resultant vector is converted into pseudorandom vector set by using a secret key called as biohash, which is unique for each user, and can be stored in a storage device for verification purposes.

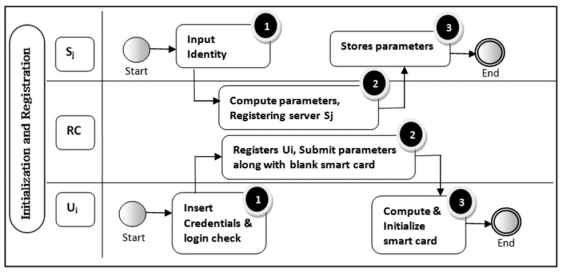

Fig. 2. General representation of user and server registration phase.

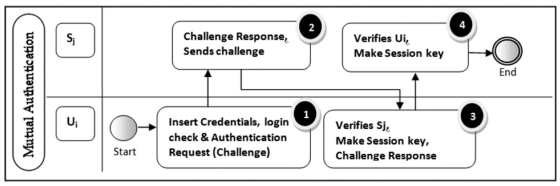

Fig. 3. General representation of Mutual Authentication phase.

III. Proposed Model

The motivation of proposed work is to design a MSA protocol which provides anonymous authenticated services to users ensuring efficient mutual authentication between user and server escaping registration authority. In view of limitations in previous studies as elaborated above, we proposed an anonymous and efficient MSA protocol employing elliptic curve-based operations bypassing costly pairing operations. Our scheme is also proved under formal random oracle model and automated analysis. The proposed model incorporates five sections, i.e., system setup, server registration phase, user registration phase, login and authentication phase, and password update phase. The general representation of our scheme is given in Figs. 2 and 3.

A. System Setup

The proposed MSA framework involves three interacting roles, i.e., user Ui, service provider Sj, and registration centre RC. The RC chooses the master secret key as r_m, the long term secret as x, and public key as $r_m P$. The secret r_m is solely held by RC, while x is shared among all service providers, and $r_m P$ is publicized. In Table I, we list few symbols as employed in the proposed scheme.

B. Server Registration Phase

In this phase, Sj is supposed to get registered through registration centre by adopting the following steps:

1) Primarily, the server Sj sends its identity ID_S toward RC for the objective of registration.
2) RC, then, chooses a random number v_j and computes $S_{jp} = v_j P$ and $S_{js} = v_j + r_m\ h(ID_S, S_{jp})$. Further, it communicates the message $\{S_{jp}, S_{js}, x\}$ toward Sj using secure channel.

Authorized licensed use limited to: Abu Dhabi University. Downloaded on March 13,2025 at 11:15:12 UTC from IEEE Xplore. Restrictions apply.

TABLE I
LIST OF SYMBOLS

Notations	*Description*
Ui, Sj, RC	i^{th} User, j^{th} Service provider, Registration Centre
ID_U, ID_S	Ui's identity, Sj's identity
PWi, *BIOi*	Ui's password and biometric mark
U_{is}, S_{js}:	Ui's and Sj's long term computed secret values
U_{ip}, S_{jp}:	Ui's public key, Sj's public key
r_m, r_mP	RC private secret, RC's public key
x_u, x_s	Temporary secrets
x	Shared secret among all servers
T_i, T_j	Current timestamps
$H_F()$	Bio-hashing function
$h()$,⊕, ||	Secure hash function, XOR, Concatenation

3) Thereafter, the server receives $\{S_{jp}, S_{js}, x\}$ and stores these parameters safely [46], [47].

C. User Registration Phase

In this section, the user Ui performs registration with RC by using the following steps for registration:

1) Initially, user selects its identity ID_U and password PWi. It also chooses random numbers as w and z, and imprints biometric $BIOi$ in biometric scanner. Then, it computes $PWF = h(h(ID_U||PWi)||w)$ and $h(PWi||w)$, and sends $\{ID_U, PWF \oplus z, h(PWi||w)\}$ to RC using secure channel.
2) RC, then, chooses a random number v_i, and computes $U_{ip} = v_iP, U_{is} = v_i + r_m.h(ID_U, U_{ip}), Q' = PWF \oplus z \oplus U_{is}, C_i = h(ID_U||x) \oplus h(ID_U||h(PWi||w)||U_{is}), PWG = h(ID_U||h(PWi||w)||U_{ip})$. RC provides computed parameters $\{U_{ip}, Q', C_i, PWG\}$ along with a blank smart card to Ui using a secure channel.
3) Ui, then, computes $E = w \oplus H_F(BIOi)$ and $Q = Q' \oplus z$ to finally store $\{U_{ip}, Q, C_i, PWG, E\}$ in the smart card provided by RC. Thus, the smart card is initialized by the user.

D. Login and Authentication Phase

In this section, Ui and Sj converge on a mutually agreed session key as shown in Fig. 2. The procedure is elaborated as under:

1) The user inputs identity (IDi), password (PWi), and imprints the biometric $(BIOi)$ using biometric scanner. Next, the smart card calculates $w = E \oplus H_F(BIOi), PWG* = h(ID_U||h(PWi||w)||U_{ip})$, and verifies $PWG* ? = PWG$. If it does not match, SC terminates the session. On the other hand, SC chooses a random integer x_u and computes x_uP and it submits $m_1 = \{x_uP\}$ toward server as shown in Fig. 4.
2) The server receives m_1 and chooses random number x_s. Next, it computes x_sP and $H_j = x_s + S_{js}\, h(x_uP, x_sP)$. Further, it submits $m2 = \{x_sP, H_j, S_{jp}\}$ toward Ui for verification.
3) Ui receives m_2 and verifies H_jP, i.e., $H_jP ? = (x_sP + S_{jp} + h\,(ID_S, S_{jp})r_mP).(h(x_uP, x_sP))$. If the equality holds, Ui calculates $PWF = h(h(ID_U||PWi)||w)$, $U_{is} = PWF \oplus Q, h(ID_U||x) = C_i \oplus h(ID_U||h(PWi||w)||U_{is})$, $H_i = x_u + U_{is}\, h(ID_S, x_uP, x_sP)$, $SKji = h(x_ux_sP||h(ID_U||x)||U_{ip}), V = h(x_ux_sP)||x_sP) \oplus (U_{ip}||ID_U||ID_S)$ and $Z = h(SKji, U_{ip}, x_uP, ID_U)$. Ultimately, it sends $m_3 = \{x_uP, H_i, V, Z\}$ toward the Sj for verification.
4) Sj receives $m_3 = \{x_uP, H_i, V, Z\}$ and calculates $(U_{ip}||ID_U||ID_S) = (h(x_sx_uP)||x_sP)) \oplus V$ and derives (x_uP, U_{ip}, ID_U). Further, it computes and verifies the equality for $x_sP' ? = x_sP$ and $H_iP ? = (x_uP + U_{ip} + h(ID_U, U_{ip})\, r_mP).(h(ID_S, x_uP, x_sP))$. If the equality holds true, the server calculates $h(ID_U||x)$ and $SKij = h(x_sx_uP||h(ID_U||x)||U_{ip})$ and compares again the match for $Z ? = h(SKij, U_{ip}, x_uP, ID_U)$. If the equality is positive, the server finally authenticates the user as a valid entity, with agreed session key $SKij$.

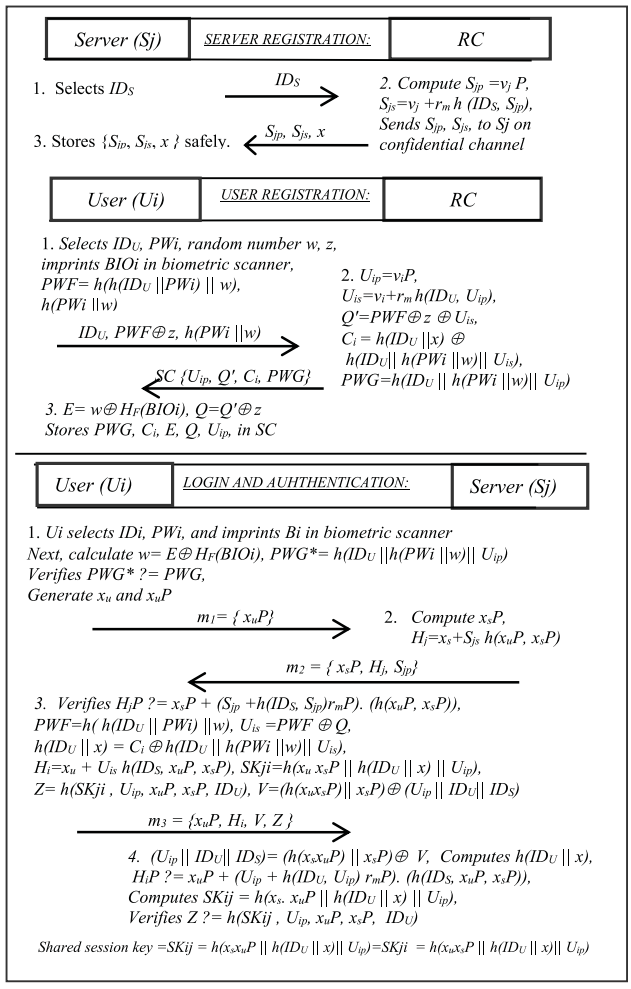

Fig. 4. Proposed scheme's registration, login, and authentication phase.

E. Password Upgrading Procedure

Ui modifies its password by initiating the following steps with server, without referring toward RC.

1) The user puts its smart card in device scanner. Then, it inputs its identity ID_U*, password $PWi*$, and imprints

Authorized licensed use limited to: Abu Dhabi University. Downloaded on March 13,2025 at 11:15:12 UTC from IEEE Xplore. Restrictions apply.

its biometric identity $BIOi*$ into biometric sensor. Then smart card calculates $w = E \oplus H(BIOi*), PWG* = h(ID_U \,||h(PWi \,||\, w) \,||\, U_{ip})$, and checks the equality for $PWG* \,? = PWG$. If it does not hold true, it terminates the session, otherwise, prompts the user to enter a new password PWi^{new}.

2) Next SC computes $PWF = h(h(ID_U||PWi)||w)$, $U_{is} = PWF \oplus Q$ and $h(ID_U||x) = C_i \oplus h(ID_U||\, h(PWi||w)||\, U_{is})$.

3) Further, it computes $PWG^{\text{new}} = h(ID_U \,||h(PWi^{\text{new}} \,||\, w) \,||\, U_{ip}), PWF' = h(h(ID_U \,||\, PWi^{\text{new}}) \,||\, w), Q^{\text{new}} = PWF' \oplus U_{is}$, and $C_i{}^{\text{new}} = h(ID_U \,||\, x) \oplus h(ID_U \,||\, h(PWi^{\text{new}} \,||w)||\, U_{is})$.
Ultimately, the parameters PWG, Ci and Q are swapped with $PWG^{\text{new}}, Ci^{\text{new}}$ and Q^{new} in smart card.

IV. Security Analysis

This section comprises formal and informal security discussion including automated tool analysis.

A. Informal Security Analysis

The informal security analysis of our scheme is as under:

1) Resists Replay Attack: An adversary $\mathcal{A}$, after accessing the intercepted parameters $\{x_uP, H_i, V, x_sP, H_j, S_{jp}, ID_S, Z\}$ might replay to betray the legal participants. Nonetheless, $\mathcal{A}$ will not be able to initiate replay attack, since, if $\mathcal{A}$ replays $m_1 = \{x_uP\}$ toward server, it may not be able to compute an updated H_i corresponding to x_sP for lacking U_{is}. Likewise, If $\mathcal{A}$ replays $m_1 = \{x_uP, H_i, V, Z\}$ toward server, the later may detect this threat by computing and verifying $H_iP\,? = (x_uP + U_{ip} + h\,(ID_U, U_{ip})\, r_m.P).\,(h(ID_S, x_uP, x_sP))$ and $Z\,? = h(SKij, U_{ip}, x_uP, x_sP, ID_U)$ in step 4 (see Fig. 2). Likewise, if $\mathcal{A}$ replays $m_2 = \{x_sP, H_j, S_{jp}\}$ to user, then the user may discern possibility of replay attack on verifying $H_jP\,? = (x_sP + S_{jp} + h\,(ID_S, S_{jp})r_m.P).(h(x_uP, x_sP))$. Hence, we can infer that the contributed scheme is protected from replay attacks.

2) Resists Offline-Password Guessing Threats: In proposed scheme, $\mathcal{A}$ might access $\{x_uP, H_i, V, x_sP, H_j, S_{jp}, ID_S, Z\}$ on scrutinizing public channel. Besides, it may also derive smart card contents $\{PWG, C_i, E, Q, U_{ip}\}$, employing side-channel attack. Nonetheless, $\mathcal{A}$ may not extract PWi of those contents, as PWi is not used in the construction of those factors except PWG and Q. While, for guessing PWi from PWG and PWF, the factor w is crucial and cannot be deduced from E unless the factor $BIOi$ is known. Thus, our protocol is immune to offline-password guessing threat.

3) Anonymous and Untraceable: In contributed scheme, Ui sends its identity ID_U in the form of computed factor V, i.e., $V = h(x_ux_sP) \,||\, x_sP) \oplus (U_{ip}, ID_U)$. Since, the factor x_ux_sP may be constructed by a legitimate server only; we can safely infer that the identity ID_U is communicated to a valid server. In this way, the ID_U remains undisclosed, and is revealed only to a legal server possessing a valid S_{js}. Besides, the exchanged messages do not comprise any factor which remains fixed in multiple sessions enabling $\mathcal{A}$ to associate any session to a particular user. Thereby, the proposed scheme warrants the anonymity or untraceability traits for its users.

4) Resists Impersonation or Masquerading Attack: The proposed protocol is resistant to server impersonation threats, given that a malicious legal server will not be able to compute a legitimate and updated H_j parameter after receiving $m_1 = \{x_uP\}$. $\mathcal{A}$ can, however, replay H_j that will be successfully detected by Ui during equality matching for $H_jP\,? = (x_sP + S_{jp} + h\,(ID_S, S_{jp})r_m.P).\,(h(x_uP, x_sP))$. Hence, it cannot generate a valid H_j by itself on intercepting m_1 and forge the user. Likewise, $\mathcal{A}$ cannot construct a legitimate H_i corresponding to x_sP in response to a valid server until $\mathcal{A}$ has access to user's secret parameters. Therefore, the proposed scheme is protected from user impersonation attack as well as server masquerading attack.

5) Resists Privileged Insider Attack: The proposed scheme may foil privileged insider attack, if a privileged insider gets access to the registration request $\{ID_U, PWF \oplus z, h(PWi||w)\}$. After intercepting those contents, $\mathcal{A}$ can neither recover previous session keys, nor initiate any impersonation attack. To construct legitimate session keys, $\mathcal{A}$ needs $(ID_U||x)$ and temporary session variables, i.e., x_u or x_s. Besides, it requires access to U_{is} to initiate user impersonation attack against a valid user. Thus, our scheme is immune to privileged insider attack.

6) No Session-Specific Temporary Information Threat: Our scheme protects the session keys, in case the session specific temporary variables are revealed to an adversary, inadvertently, i.e., x_u or x_s. $\mathcal{A}$ also needs access to user's dynamic key $h(ID_U||x)$ to construct a valid session key $SKji = h(x_u\, x_sP||h(ID_U||x)||U_{ip})$, whereas $h(ID_U \,||x)$ cannot be extracted from C_i in smart card unless the values $h(PWi||w)$ and U_{is} are known. Thus, our protocol is safe from session-specific temporary information threat.

7) No Key-Compromise Impersonation Attack (KCI): The KCI attack is possible on a scheme if some revealed parameter of user makes the adversary initiate a successful server masquerading attack. In proposed model, despite the disclosure of user's parameter U_{is}, $\mathcal{A}$ cannot launch a server masquerading attack, since $\mathcal{A}$ will not be able to manufacture the fresh $H_j = x_s + S_{js}h(x_uP, x_sP)$ without sufficient information regarding the server' key S_{js}. A legitimate user could thwart this attack by checking the equality for equation $H_jP\,? = (x_sP + S_{jp} + h(ID_S, S_{jp})r_m.P).\,(h(x_uP, x_sP))$. Therefore, the proposed model is resistant to KCI attack.

B. Automated Tool Analysis Using ProVerif

We employed the ProVerif automated tool [43], [44] for proving the subtle security features of our protocol. We initiate the protocol testing procedure in ProVerif tool after setting two communication channels (SCh as private and PCh as public) among participants, i.e., RC, Ui, and Sj.

We define the two beginning and ending events for each member, Ui as *beginUserUi(bitstring)* & *endUserUi(bitstring)*, and Sj as *beginServerSj (bitstring)* & *endServerSj(bitstring)*, as shown in Fig. 5.

Authorized licensed use limited to: Abu Dhabi University. Downloaded on March 13,2025 at 11:15:12 UTC from IEEE Xplore. Restrictions apply.

```
(***************** Channels *****************)
free SCh:channel [private].       (*Secure Channel*)
free PCh:channel.            (*Public Channel*)
(********** Constants & Variables **********)
const P: bitstring .
free IDu : bitstring.
free IDs : bitstring.
free rm : bitstring [ private ] .
free X : bitstring [ private ] .
(*********** Constructor ***********)
fun h( bitstring ) : bitstring .
fun H( bitstring ) : bitstring .
fun ECPM( bitstring , bitstring ) : bitstring .
fun ENC(bitstring,bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.
fun MULT( bitstring , bitstring ) : bitstring .
fun ADD( bitstring , bitstring ) : bitstring .
(*fun CONCAT( bitstring , bitstring ) : bitstring .*)
(*********** Destructors & Equations ***********)
equation forall a:bitstring,b:bitstring; XOR(XOR(a,b),b)=a.
reduc forall m:bitstring,key:bitstring; DEC(ENC(m, key), key)=m.
(***************** Events *****************)
event beginUserUi ( bitstring ) .
event endUserUi ( bitstring ) .
event beginServerSj ( bitstring ) .
event endServerSj ( bitstring ) .
```

Fig. 5. Initialization code for simulation.

```
(********************* p r o c e s s e s *********************)
( ***************** User Ui ***************** )
let UserUi=
( ****** Registration ****** )
new w: bitstring;
new z: bitstring;
let PWF= h(CONCAT(h(CONCAT(IDu, PWi)), w)) in
let Pw=h(CONCAT(PWi, w)) in
let PWF''=XOR(PWF, z) in
out (SCh , (IDu, PWF'', Pw)) ;
in (SCh, (xUip : bitstring, xQ':bitstring, xCi:bitstring , xPWG:bitstring,)) ;
let E=XOR(w, H(Bi)) in
let Q=XOR(z, xQ') in

( ******** Login and Authentication ********* )
event beginUserUi ( IDu ) ;
let w= XOR(E, H(Bi)) in
let PWG'= h(CONCAT(IDu, h(PWi, w))) in
if PWG'= xPWG ) then
new xu:bitstring;
let XuP = ECPM(xu ,P) in
out (PCh, (XuP)) ;
    in (PCh, (xXsP: bitstring, xHj: bitstring, xxSjp: bitstring, xIDs: bitstring)) ;
let HjP = ECPM( xHj ,P) in
let HjP' = ADD(xXsP, MULT (ADD(xSjp , MULT(h( xIDs , xSjp ), ECPM(rm, P))) , h(XuP, xXsP))) in
if (HjP = HjP') then
let PWF= h(CONCAT(h(CONCAT(IDu, PWi)), w)) in
let Uis = XOR(J, PWF) in
let PWN= h(CONCAT(IDu, h(PWi, w), Uis)) in
let Ci' = XOR(xCi, PWN) in
let Hi = ADD(xu, MULT( Uis ,h(IDs, XuP, xXsP))) in
let SKji = h(h(ECPM(xu, xXsP), Ci', xUip) in
let V = XOR(CONCAT(h(ECPM(xu, xXsP)), xXsP), CONCAT(xUip, IDu, IDs)) in
let Z=h(SKji, xUip, XuP, xXsP, IDu)
out (PCh, (XuP, Hi, V, Z)) ;
event endUserUi (IDu)
else
0.
```

Fig. 6. Simulation code for UserUi process.

We portray three separate processes $Registration\ CentreRC, UserUi$, and $ServerSj$ against RC, Ui, and Sj, respectively. The $UserUi$ initially sends (IDu, PWF", Pw) and receives (xU_{ip}, xQ, xCi, xPWG) using SCh. In mutual authentication phase, $UserUi$ sends (XuP, Hi, V, Z) and receives xXsP, xHj, xSjp, xIDs from *ServerSj* using PCh. Finally, it submits XuP, Hi, V, Z to *ServerSj*. From *UserUi, RegistrationCentreRC* receives xIDu and submits (U_{ip}, Q', Ci, PWG) to it using SCh. Similarly, to register a server it receives xIDs from *ServerSj* and submits (S_{jp}, S_{js}, X) to the same process using SCh, as shown in Fig. 6.

The *ServerSj* sends IDs and receives xS_{jp}, xS_{js}, xX from *RegistrationCentreRC* using SCh to finalize registration. In mutual authentication, *ServerSj* receives xXuP, xIDs from *UserUi*, computes XsP, Hj, and sends (XsP, Hj, xSjp, IDs) to *UserUi*. Next, it receives (xXuP, xHi, xV, and xZ) from *UserUi*. It decrypts xV and gets xxXsP, xxU_{ip}, and xIDu. Then, it computes Ci", HiP, and HiP' and checks its validity. If positive, *ServerSj* approves *UserUi* and the constructed session key, i.e., SKij, as shown in Fig. 7.

```
(****************** Registration Centre (RC) ******************)
let RegistrationCentreRC =
( ******* User Registration ********* )
in (SCh , xIDu : bitstring, xPWF": bitstring, xPw: bitstring ) ;
new vi : bitstring ;
let Uip = ECPM(vi ,P) in
let Uis = ADD(vi , MULT(rm ,h(xIDu, Uip ))) in
let Q'= XOR(xPWF", Uis) in
let Ci'= h(CONCAT(IDu, X)) in
let PWN= h(CONCAT(xIDu, xPw, Uis) in
let Ci = XOR(Ci', PWN) in
let PWG=h( CONCAT(IDu , xPw , Uip)) in
out (SCh, (Uip , Q', Ci, PWG)) ;
( ******* Server Registration ********* )
in (SCh , xIDs : bitstring ) ;
new vj : bitstring ;
let Sjp = ECPM( vj ,P) in
let Sjs = ADD( vj ,MULT( rm , h( xIDs , Sjp ) ) ) in
out (SCh , ( Sjp , Sjs, X )) ;
0 .
   (****************** Server (Sj) ******************)
( * Server Sj * )
let ServerSj=
( ********* Registration * )
out (SCh , IDs ) ;
in (SCh , ( xSjp : bitstring , xSjs : bitstring, xX : bitstring )) ;

( ********** Login and Authentication ********** )
event beginServerSj ( IDs ) ;
in (PCh, (xXuP)) ;
new xs:bitstring;
let XsP= ECPM( xs ,P) in
let Hj = ADD( xs ,MULT(xSjs ,h(xXuP, XsP))) in
out (PCh, (XsP, Hj, xSjp)) ;
in (PCh,(xXuP: bitstring, xHi: bitstring, xV: bitstring, xZ: bitstring )) ;
let (xxUip:bitstring, xIDu:bitstring, xIDs:bitstring)= XOR(xV, CONCAT(h(ECPM(xs,
xXuP)), XsP)) in
let Ci"= h(CONCAT(xIDu, X)) in
let HiP ' = (ADD(xXuP, MULT (ADD(xxUip, MULT(h( xIDu, xxUip) , ECPM(rm ,P)) ) , h
(IDs, xXuP, XsP)))) in
if (HiP = HiP ' ) then
let SKij = h(ECPM(xs, xXuP), Ci", xxUip) in
let Z'=h(SKij, xxUip, xXuP, XsP, xIDu) in
if (Z' = xZ) then
event endServerSj ( IDs )
else
0 .
```

Fig. 7. Simulation code for RegistrationCentreRC and ServerSj.

```
(******************** Queries *********************)
free SK: bitstring [ private ] .
query attacker (SK) .
query id : bitstring ; inj event ( endUserUi ( id ) ) ==> inj
event ( beginUserUi ( id ) ) .
query id : bitstring ; inj event ( endServerSj ( id ) ) ==> inj
event ( beginServerSj ( id ) ) .
```

Fig. 8. Simulation code for queries.

The three participants interact for an infinite number of sessions. In this way these processes work in replication as depicted here.

process
((! UserUi) | (! RegistrationCentreRC) | (! ServerSj))

We design the following queries for testing security and correctness of the contributed scheme, as shown in Fig. 8.

The following three results have been obtained after the execution of these queries in the simulation.

RESULT inj-event (endServerSj(id)) $==>$ inj-event (beginServerSj(id)) is true.
RESULT inj-event (endUserUi(id_1153)) $==>$inj-event (beginUserUi(id_1153)) is true.
RESULT not attacker (SK[]) is true.

The above results demonstrate that the three processes began and terminated effectively, also the attacker query is unable to recover the session key constructed by the involved processes during mutual authentication phase.

Authorized licensed use limited to: Abu Dhabi University. Downloaded on March 13,2025 at 11:15:12 UTC from IEEE Xplore. Restrictions apply.

TABLE II
COMPUTATIONAL COST COMPARISON

Execution ($\prod_{\mathrm{Ui}}^{\psi}$, $\prod_{\mathrm{Sj}}^{\partial}$): This query could be used by adversary to launch passive attacks after eavesdropping executions between $\prod_{\mathrm{Ui}}^{\psi}$ and $\prod_{\mathrm{Sj}}^{\partial}$.
SendClient ($\prod_{\mathrm{Ui}}^{\psi}$, m): This query is launched by $\mathcal{A}$ to intercept and submit messages to $\prod_{\mathrm{Ui}}^{\psi}$ after modifying it.
SendServer($\prod_{\mathrm{Sj}}^{\partial}$, m): This query is used by $\mathcal{A}$ to initiate an active attack on Sj. Its output reveals message as generated by $\prod_{\mathrm{Sj}}^{\partial}$ after receiving m.
Reveal ($\prod_{\mathrm{Ui}}^{\psi}$): $\mathcal{A}$ utilize Reveal query for initiating known-session key attack, and extracting session key for corresponding $\prod_{\mathrm{Ui}}^{\psi}$.
Corrupt (*Ui*): This query returns to the adversary, the long term secret (*PWi)* of the interacting participant Ui.
Test ($\prod_{\mathrm{Ui}}^{\psi}$) : $\mathcal{A}$ sends a single Test query to the fresh oracle. In return, this oracle instance chooses at random a bit $\mathscr{b} \in \{0, 1\}$ and returns the corresponding session key, if $\mathscr{b}$=1. Or else, it will return a random value in the response out of {0, 1}*.

C. Formal Security Analysis

We demonstrate the formal security analysis using Abdalla and Pointcheval model [39] in order to verify the resilience of security properties of contributed model.

1) Participating Roles: In our protocol $\prod$, the participating entities such as user Ui $\in \mathcal{U}$ and server Sj $\in \mathcal{S}$, cooperate in a network. Every participant includes multiple instances (oracles) in each implementation of $\prod$. We symbolize an ψth instance of user as $\prod_{\mathrm{Ui}}^{\psi}$ and ∂th instance of server as $\prod_{\mathrm{Sj}}^{\partial}$ for a session. Each of instances, i.e., $\prod_{\mathrm{Ui}}^{\psi}$ or $\prod_{\mathrm{Sj}}^{\partial}$ bears a partnering-based identity $\mathcal{P}_{id_{Ui}}^{\psi}$ or $\mathcal{P}_{id_{Sj}}^{\partial}$, session-based identity $S_{id_{Ui}}^{\psi}$ or $S_{id_{Sj}}^{\partial}$, and session key $\mathscr{s}k_{Ui}^{\psi}$ or $\mathscr{s}k_{Sj}^{\partial}$, respectively. The $\mathcal{P}_{id_{Ui}}^{\psi}$ or $\mathcal{P}_{id_{Sj}}^{\partial}$ identities portray a set of identities associated with a particular instance. The identities,$S_{id_{Ui}}^{\psi}$ or $S_{id_{Sj}}^{\partial}$ represent session-specific streams or flows related to any specific instance. Likewise, an instance such as $\prod_{\mathrm{Ui}}^{\psi}$ or $\prod_{\mathrm{Sj}}^{\partial}$ is considered as *accepted*, if it bears the corresponding session key $\mathscr{s}k_{Ui}^{\psi}$ or $\mathscr{s}k_{Sj}^{\partial}$. These instances are treated as partnered, if both of these instances initially, and the equality for $\mathcal{P}_{id_{Ui}}^{\psi}$ and $\mathcal{P}_{id_{Sj}}^{\partial}$, $S_{id_{Ui}}^{\psi}$, and $S_{id_{Sj}}^{\partial}$, and also for $\mathscr{s}k_{Ui}^{\psi}$ and $\mathscr{s}k_{Sj}^{\partial}$ holds at the same time.

2) Long-Term Secrets: The user Ui $\in \mathcal{U}$ selects password (*PWi)* and a biometric (*BIOi)* for user's smart card verification, where for the purpose of authentication it uses the computed U_{ip} and U_{is}, as received from RC. Likewise, the server Sj $\in \mathcal{S}$ is issued S_{jp} and S_{js} from RC. These parameters are used by Ui and Sj to mutually authenticate one another.

3) Adversarial Model: In this adversarial model, we take on some assumptions associated with the capabilities of an attacker corresponding to our protocol, as given below:

1) We assume that $\mathcal{A}$, a probabilistic polynomial time attacker with the capability of managing communication means could access, insert, alter, and obstruct the exchanged messages on public communication channels.
2) $\mathcal{A}$ may interact with any legal entity by using oracle queries and develop the capability of initiating an attack.

$\mathcal{A}$ might issue the queries in an order given in Table II.

Definition 4: Fresh Oracle

The oracle $\prod_{\mathrm{Ui}}^{\psi}$ is termed as *fresh*, provided 1) $\prod_{\mathrm{Ui}}^{\psi}$ lies in *accepted* state, or otherwise, $\prod_{\mathrm{Ui}}^{\psi}$ and the corresponding partner constructs an agreed session key. 2) After the acceptance, *Reveal* queries are not again submitted to $\prod_{\mathrm{Ui}}^{\psi}$ or its partner.

Definition 5: Protocol's strength

Here, we simulate the protocol ($\prod$)'s strength by employing a game *Game*($\prod, \mathcal{A}$). Where, the adversary may launch *SendClient, SendServer*, *Execution*, *Corrupt*, and *Reveal* queries, nonetheless, it could initiate the *Test* query just one time to a particular oracle. If the attacker initiates *Test* query on getting accepted of the oracle $\prod_{\mathrm{Ui}}^{\psi}$, the adversary will yield a single bit $\mathscr{b}'$ The aim of the attacker is to deduce that bit $\mathscr{b}$ appropriately in the test session. Now, the attacker's advantage can be shown as

$$Adv_{\mathcal{A}}^{\mathrm{APAMCC}}(\mathcal{A}) = \left| 2\Pr\left(\mathscr{b}' = \mathscr{b}\right) - 1 \right|. \tag{4}$$

Our APAMCC protocol is regarded as secure, if advantage ($Adv_{\mathcal{A}}^{\mathrm{NPAMC}}$) of $\mathcal{A}$ is considerably negligible.

Subsequently, for verifying the security properties of our protocol, we utilize the DDH assumption as given below:

Definition 6: Diffie—Helman (DDH) assumption

This property can be illustrated by two experiments, $Exp_{P,g}^{\mathrm{ddh-real}}(\mathbb{A})$ and $Exp_{P,g}^{\mathrm{ddh-rand}}(\mathbb{A})$. Here, the adversary $\mathbb{A}$ is presented with $x_u P, x_s P$, and $x_u x_s P$ in experiment $Exp_{P,g}^{\mathrm{ddh-real}}(\mathbb{A})$, and $x_u P, x_s P$, and $x_d P$ in $Exp_{P,g}^{\mathrm{ddh-rand}}(\mathbb{A})$. Where, x_u, x_s, and x_d are selected at random out of Z_n^*. Where, g represents a large prime integer and P be a generator of a finite cyclic group 'G'. The $\mathbb{A}$'s advantage for violating DDH assumption is

$$Adv_{P,g}^{\mathrm{ddh}}(\mathbb{A}) = \max\{|Pr[Exp_{P,g}^{\mathrm{ddh-real}}(\mathbb{A}) = 1] - \Pr[Exp_{P,g}^{\mathrm{ddh-rand}}(\mathbb{A}) = 1]|\}. \tag{5}$$

4) Authentication Proof:

Theorem 1: We represent a symbol $\mathfrak{D}$ for an equally distributed password dictionary with size $|\mathfrak{D}|$, and Π as the protocol. Hence, as per definition of DDH, we say

$$Adv_{\mathfrak{D},\Pi}(\mathcal{A}) \leq \frac{qr_h^2}{2^l} + \frac{(q_{r_s} + q_{r_e})^2}{g^2} + 2qr_e.\, Adv_{P,g}^{\mathrm{DDH}}(\mathbb{A}) + 2\max\left\{\frac{qr_h}{2^k}, \frac{qr_s}{|\mathfrak{D}|}\right\} \tag{6}$$

where the parameters $qr_e, qr_h,$ and qr_s depict the number of *Execution*, *Hash*, and *SendClient* queries, respectively. Where, l specifies the length of user's identity and password, and g describes a high entropy prime integer.

Theorem: proof (Game attacks)

We shall prove this theorem with a series of attack games that begins from a genuine attack G_0 and terminating at G_4, where $\mathcal{A}$ will not be having significant advantage. We may delineate an event ε_i for each game G_i where i range from $\{0 \leq i \leq 4\}$, so that the adversary suitably guesses bit $\mathscr{b}$ in the *Test* query.

Game G_0*:* The game G_0 is an actual attack or protocol in the ROM model, while all of the instances for Ui and Sj are modeled as factual implementation in the oracle. For the event

Authorized licensed use limited to: Abu Dhabi University. Downloaded on March 13,2025 at 11:15:12 UTC from IEEE Xplore. Restrictions apply.

ε_0, it can be inferred that $\mathcal{A}$ effectively made the guess of bit $\mathscr{b}$ in *Test* query, i.e.,

$$Adv_{\mathfrak{D},\prod}\ \ (\mathcal{A}) = \left|2Pr\left[\varepsilon_{v0}\right] - \frac{1}{2}\right|. \tag{7}$$

Game G_1: This game is alike G_0 game, nonetheless it models hash-based oracles h and H by using hash list L_h bearing entries (*In*, *Op*). This list responds as *Op* after going through (*In*, *Op*) on getting hash query. Otherwise, it chooses *Op* randomly, modifies the corresponding entry (*In*, *Op*), and sends to the adversary. From $\mathcal{A}$'s outlook, the actual implementation of the real attack (G_0) on the protocol and G_1 game's simulation are not possible to differentiate. As the oracle-based queries, i.e., Test, Reveal, Hash, Send, Execute, and Corrupt are modeled equally as a genuine attack. Thereby

$$\Pr\left[\varepsilon_{v1}\right] = \Pr\left[\varepsilon_{v0}\right]. \tag{8}$$

Game G_2: The Game G_2 is alike game G_1 and we perform simulation for all oracles except that if $\mathcal{A}$ guesses secrets of the involved participants, given that the partial transcripts $\{x_u P, x_s P\}$ find a match with real values or face collisions, the executions are simply aborted. Referring to birthday paradox, the probability chance for collisions in random output of hash oracles is not higher than $\frac{qr_h^2}{2^{l+1}}$. Likewise, its chances in those transcripts are at most $\frac{(q_{r_s}+q_{r_e})^2}{2g^2}$, while qr_e and qr_s depict the number of *Execution* and *SendClient* queries, respectively. Thus

$$\left|\Pr\left[\varepsilon_{v2}\right] - \Pr\left[\varepsilon_{v1}\right]\right| \le \frac{qr_h^2}{2^{l+1}} + \frac{(q_{r_s}+q_{r_e})^2}{2g^2}. \tag{9}$$

Game G_3: In this game, simulation of the queries is updated as SendClient oracle. Initially, we select a session at random, executed by partnered instances $\prod_{\mathrm{Ui}}^{\psi}$ and $\prod_{\mathrm{Sj}}^{\partial}$

1) Upon receiving SendClient ($\prod_{\mathrm{Ui}}^{\psi}$, Start), we select at random, a value u' ϵ Z_n^* to compute $x_{u'}P$ and submit to $\mathcal{A}$.
2) Upon receiving SendServer($\prod_{\mathrm{Sj}}^{\partial}$, $x_{u'}P$) query, we select $s' \epsilon Z_n^*$ and compute $x_{s'}P, Hj, x_{s'}x_{u'}P, S_{jp}$ similar to real authentication protocol, and return $\{x_{s'}P, Hj, S_{jp}\}$ to $\mathcal{A}$.
3) Next, upon receipt of SendClient($\prod_{\mathrm{Ui}}^{\psi}$, $(x_{s'}P, Hj, S_{jp})$), we compute $x_{u'}P, Hi, V, Z$ and return $\{x_{u'}P, Hi, V, Z\}$ to $\mathcal{A}$.

Therefore, it is noticeable that G_3 is improbable to differentiate from earlier game G_2. Therefore

$$\Pr\left[\varepsilon_{v3}\right] = \Pr\left[\varepsilon_{v2}\right]. \tag{10}$$

Game G_4: In G_4, the simulation for queries is again altered to SendClient oracle for a specific session in game G_3. For this, the procedure of computing $x_s x_u P$ is modified in order to minimize dependence on password and temporary parameters. When SendServer ($\prod_{\mathrm{Sj}}^{\partial}$, $x_u P$) and SendClient($\prod_{\mathrm{Ui}}^{\psi}$, $(x_s P, Hj, S_{jp})$) are requested, we compute $x_d P$, where x_d is randomly chosen from Z_n^*. The distinction between G_3 and G_4 becomes obvious from the following equation:

$$\left|\Pr\left[\varepsilon_{v4}\right] - \Pr\left[\varepsilon_{v3}\right]\right| \le q_{r_e} . Adv_{P,g}^{\mathrm{DDH}}\ (\mathcal{A}). \tag{11}$$

Following this equation, $\mathcal{A}$ may design a DDH solver by distinguishing G_4 and G_3. In G_4, we employed a randomly selected Diffie–Hellman key x_d, not depending on PWi or other variables. The random key x_d might be distinguished from a legal session key (SK) in three ways, i.e.,

Case 1: In case 1, $\mathcal{A}$ puts a query $Z = h(SKij, U_{ip}, x_u P, x_s P, ID_U)$ to h. The probability of happening for the said event is $\frac{qr_h}{2^l}$.

Case 2: $\mathcal{A}$ requests SendClient query except SendClient ($\prod_{\mathrm{Sj}}^{\partial}$, m) and successfully masquerades Ui as Sj. The attacker is not sanctioned for approaching the static key such as PWi of Ui. Hence, $\mathcal{A}$ needs to acquire PWi knowledge of Ui for initiating a forgery attack. Its probability is calculated as $1/\mathfrak{D}$. As, there are at most (qr_s) number of sessions of that type, the associated probability of that event are quite less than $qr_s/|\mathfrak{D}|$. Thus, we infer

$$Pr\left[\varepsilon_{v4}\right] = \frac{1}{2} + \max\left\{\frac{qr_h}{2^k}, \frac{qr_s}{|\mathfrak{D}|}\right\}. \tag{12}$$

Using (6)–(12), one may conclude as

$$\Pr\left[\varepsilon_{v4}\right] = \frac{1}{2} + \max\left\{\frac{qr_h}{2^k}, \frac{qr_s}{|\mathfrak{D}|}\right\}$$

$$Adv_{\mathfrak{D},\prod}\ \ (\mathcal{A}) = 2\left|\Pr\left[\varepsilon_{v0}\right] - \frac{1}{2}\right|$$

$$= 2\left(\left|\Pr\left[\varepsilon_{v0}\right] - \Pr\left[\varepsilon_{v4}\right]\right| + \max\left\{\frac{qr_h}{2^k}, \frac{qr_s}{|\mathfrak{D}|}\right\}\right)$$

$$\le 2\Bigg(\left|\Pr\left[\varepsilon_{v1}\right] - \Pr\left[\varepsilon_{v2}\right]\right| + \left|\Pr\left[\varepsilon_{v3}\right] - \Pr\left[\varepsilon_{v4}\right]\right| + \max\left\{\frac{qr_h}{2^k}, \frac{qr_s}{|\mathfrak{D}|}\right\}\Bigg) \le \frac{qr_h^2}{2^l}\ \frac{(q_{r_s}+q_{r_e})^2}{g^2}$$

$$+\ 2q_{r_e} \cdot\ Adv_{P,g}^{DDH}\ (\mathcal{A})\ 2\max\left\{\frac{qr_h}{2^k}, \frac{qr_s}{|\mathfrak{D}|}\right\}\Bigg).$$

V. Performance Evaluation

In this section, we investigate and evaluate the performance of proposed model against other MRC-Off schemes. Table III shows the security features' comparison of various schemes, which suggests the proposed scheme as robust in comparison with other MRC-Off schemes. In Table III, the first seven schemes in comparison relate to SSSS category, the subsequent three protocols relate to SSSC type, the next scheme in the line is SUSP type, while the last two specify SCBP type, respectively. The protocols lying under the same category in Table III, share the same weaknesses in general. It is obvious in Table III that SSSS category protocols are prone to server impersonation threat. The SSSC and SUSP category protocols are unscalable for bearing the assumption that either number of servers or users is fixed in the network. The SCBP schemes are designed using costly bilinear operations. The proposed protocol is protected of all of the specified flaws as discovered in previous schemes also shown in Table III.

The Table V depicts the computational costs of various schemes as categorized separately. We can see in Table V that SSSS-type schemes [28]–[33] have very low computational cost bearing hash-based cryptographic operation except [34];

Authorized licensed use limited to: Abu Dhabi University. Downloaded on March 13,2025 at 11:15:12 UTC from IEEE Xplore. Restrictions apply.

TABLE III
SECURITY FEATURES OF MSA PROTOCOLS WITH OFFLINE RC

Schemes	[28]	[29]	[30]	[31]	[32]	[33]	[34]	[35]	[36]	[37]	[38]	[46]	[47]	**Ours**
F1	✓	✓	✓	✓	✓	×	✓	×	✓	×	✓	×	✓	✓
F2	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓
F3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓
F4	×	×	✓	✓	✓	✓	✓	×	✓	×	✓	✓	×	✓
F5	✓	✓	✓	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓
F6	✓	✓	✓	×	✓	✓	✓	✓	×	✓	✓	✓	✓	✓
F7	✓	×	×	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓
F8	×	×	×	✓	✓	×	×	×	✓	×	✓	✓	✓	✓
F9	×	×	×	×	×	×	×	×	✓	×	✓	×	×	✓
F10	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	×	✓	✓	✓
F11	✓	✓	✓	✓	✓	✓	✓	×	×	×	✓	✓	✓	✓
F12	×	×	×	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
F13	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
F14	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓
F15	✓	✓	×	✓	✓	✓	✓	×	✓	✓	×	×	×	✓

F1:Anonymity, F2: Resist Replay threat, F3: Resist Denial-of-service threat, F4: Resist Offline-password guessing threat, F5: Resist Stolen smart card threat, F6: Resist Trace threat, F7: Perfect Forward Secrecy, F8: Mutual authentication, F9: Resist Impersonation threat, F10: No need to submit users-based identities to all servers, F11: No Public keys storage in Smart card during registration, F12: No Single secret sharing with all servers leading to server masquerading attack (insider), F13: No time synchronization required, F14: Pairing free protocol, F15: Resists Session-specific temporary information attack $\surd$: Resistant to Attack or compatible with feature, $\times$: Not Resistant to Attack or not compatible with the feature.

TABLE IV
COMPUTATIONAL COST COMPARISON

	User (ms)	*Server (ms)*
T_{MTP}	*29.432*	*5.387*
T_{BP}	*28.692*	*5.318*
T_{ECM}	*11.227*	*2.025*
T_{PA}	*0.078*	*0.023*
T_{Exp}	*2.351*	*0.315*
T_M	*0.012*	*0.002*
T_H	*0.067*	*0.009*
T_{SYM}	*0.134*	*0.018*
$T_{ASE/ADE}$	*411.8*	*3.7*

TABLE V
COMPUTATIONAL COST COMPARISON

Schemes		Authentication phase (Computational cost)	
		User Ui	*Server Sj*
SSSS	[28]	$8T_H \approx 0.536$	$8T_H \approx 0.072$
	[29]	$9T_H \approx 0.603$	$7T_H \approx 0.063$
	[30]	$10T_H \approx 0.67$	$7T_H \approx 0.063$
	[31]	$11T_H \approx 0.737$	$8T_H \approx 0.072$
	[32]	$8T_H \approx 0.536$	$7T_H \approx 0.063$
	[33]	$8T_H \approx 0.536$	$8T_H \approx 0.072$
	[34]	$4T_H+1T_{ASE} \approx 411.26$	$4T_H+1T_{ASD} \approx 3.776$
SSSC	[35]	$4T_H+4T_{Exp}+1T_M \approx 9.684$	$3T_H+3T_{Exp}+1T_M \approx 0.971$
	[36]	$5T_H+2T_{SYM}+2T_M \approx 0.627$	$3T_H+2T_{SYM}+2T_M \approx 0.065$
	[37]	$4T_H+2T_{Exp}+1T_M \approx 4.982$	$5T_H+4T_{Exp}+1T_M \approx 1.33$
SUSP	[38]	$5T_H+4T_{ECM} \approx 45.24$	$4T_H+4T_{ECM} \approx 8.13$
SCBP	[46]	$3T_{ECM} + 4T_H + 1T_{MTP} + 1T_{PA} + 1T_{Exp} \approx 65.81$	$2T_{ECM} + 5T_H + 2T_{BP} + 2T_{PA} + 1T_{Exp} \approx 15.091$
	[47]	$3T_{ECM} + 4T_H + 1T_{MTP} + 1T_{PA} \approx 63.461$	$5T_H + 2T_{Exp} + 2T_{BP} + 1T_{PA} \approx 11.332$
Ours	-	$4T_{ECM} + 9\ T_H + 2T_{PA} \approx 45.667$	$4T_{ECM} + 7T_H + 2T_{PA} \approx 8.209$

however have many weaknesses including server masquerading attack. The SSSC-type schemes [35]–[37] have not much computational cost; though require much memory in smart cards to store the servers-based parameters or public keys. Likewise, the SUSP-type schemes [38] have low cost as compared to proposed scheme; however vulnerable to several attacks and have scalability concerns as defined above. Besides, the SCBP schemes [46], [47] incur high computational cost due to the bilinear operations involved. Our proposed scheme may lie in SSSS type, since it shares a single secret x among all service providers, however it does not share the limitations with the schemes lying in SSSS type.

We used few notations for approximate execution timings of respective cryptographic operations, i.e., T_H be the approximate time for executing one-way hash, T_M the time for modular multiplication, T_{SYM} the time for symmetric key operation, T_{ECM} the time for elliptic curve based scalar point multiplication, T_{BP} the time for bilinear pairing operation, T_{PA} the time for point addition, T_{MTP} the time for map-to-point function, T_{Exp} the time for modular exponentiation operation, and T_{ASE} the time for asymmetric encryption or decryption.

We computed the above computational costs using MIRACL library [50] after implementing it on a mobile device (Lenovo Zuk Z1 with Quad-core 2.5 Ghz processor having 3 GB of memory and Android Operating System V5.1.1), and a personal computer (HP E8300 Core i5, 2.93 Ghz processor with 4 GB of memory using Ubuntu 16.10 Operating system). The experiments were conducted on mobile device and personal computer for user and server, respectively. The computational cost and running time of various comparative studies is depicted in Table V. The contributed protocol is resilient to all threats and pointed weaknesses that previous MRC-Off schemes were found to be involved. The proposed scheme bears less computational cost on user's end particularly as compared to the schemes in [34], [46], and [47], while on server's end it is efficient as compared to the schemes in [46] and [47]. The computational cost of our scheme is almost equivalent to [38] as evident from Table V. Although, our scheme bears somehow more computational cost than its other counterparts, it is resistant to numerous attacks and limitations of those schemes as depicted in Table III.

Table VI shows communication overhead in terms of transmitted messages and bits for different protocols in MRC-Off. The communication cost of different schemes is assumed for user identity as 160-bits, hash function (SHA-1) as 160-bits, random integer x as 160-bits, elliptic curve-based point (xP, yP) as 320-bits, bilinear-pairing operation as 320-bits, timestamp as 32-bits, digital signature (modulo exponential) as 1024-bits, and AES encryption as 256-bits [43].

Authorized licensed use limited to: Abu Dhabi University. Downloaded on March 13,2025 at 11:15:12 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
COMMUNICATION OVERHEAD

	Communication overhead (bits)
[28]	(3 message round-tips) 1120 bits
[29]	(3 message round-tips) 1120 bits
[30]	(3 message round-tips) 1280 bits
[31]	(3 message round-tips) 1280 bits
[32]	(3 message round-tips) 960 bits
[33]	(3 message round-tips) 1280 bits
[34]	(3 message round-tips) 1664 bits
[35]	(3 message round-tips) 5056 bits
[36]	(3 message round-tips) 608 bits
[37]	(3 message round-tips) 5056 bits
[38]	(2 message round-tips) 1184 bits
[46]	(2 message round-tips) 1760 bits
[47]	(2 message round-tips) 1440 bits
Ours	**(3 message round-tips) 2432 bits**

In proposed scheme, we employed SHA-256, i.e., 256-bits for hash function, and 128-bits for user or server identity. The rest of the parameters are taken as assumed. Our scheme bears communication cost, i.e., 2432 bits due to SHA-256 hash digest, which is higher than most of the schemes except [35] and [37] having 5056 bits cost due to exponential operations. Despite a little higher communication cost, the proposed scheme's security strength is duly verified using formal analysis based on random oracle model. Hence, our protocol securely and efficiently achieves mutual authentication with offline registration authority, as contrary other MSA schemes.

VI. CONCLUSION

In this protocol, we analyzed state of the art MSA protocols that authenticate user and service provider without engaging online registration centre. We discussed that the current MSA protocols, allowing mutual authentication with offline registration authority, inevitably bear one of the four drawbacks, i.e., these protocols either store servers' parameters in smart cards, or submit user-based identities to servers, or share a single key with all servers, or otherwise the solutions are based on costly pairing operations. In view of these problems, we propose an efficient MSA protocol with offline registration authority employing elliptic curve based operations that counters the limitations of contemporary multiserver protocols efficiently. Our scheme also incorporates formal security analysis employing random oracle model that warrants the strength and efficiency of the contributed scheme.

REFERENCES

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

[2] C. C. Chang and T. C. Wu, "Remote password authentication with smart cards," in *IEE Proc. E, Comput. Digit. Techn.*, vol. 138, pp. 165–168, 1991.

[3] L. H. Li, I. 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.

[4] I. C. Lin, M. S. Hwang, and L. H. Li, "A new remote user authentication scheme for multi-server architecture," *Future Gener. Comput. Syst.*, vol. 1, no. 19, pp. 13–22, 2003.

[5] Y. Sutcu, H. T. Sencar, and N. Memon, "A secure biometric authentication scheme based on robust hashing," in *Proc. 7th Workshop Multimedia Security*, Aug. 2005, pp. 111–116.

[6] C. L. Hsu, "Security of Chien et al.'s remote user authentication scheme using smart cards," *Comput. Standards Interfaces*, vol. 26, no. 3, pp. 167–169, 2004.

[7] A. Irshad, M. Sher, E. Rehman, S. A. Ch, M. U. Hassan, and A. Ghani, "A single round-trip SIP authentication scheme for VOIP using smart card," *Multimedia Tools Appl.*, vol. 74, no. 11, pp. 3967–3984, 2005.

[8] A. Irshad, M. Sher, M. S. Faisal, A. Ghani, Ul Hassan M., and S. Ashraf Ch, "A secure authentication scheme for session initiation protocol by using ECC on the basis of the Tang and Liu scheme," *Security Commun. Netw.*, vol. 7, no. 8, pp. 1210–1218, 2014.

[9] S. A. Chaudhry, "A secure biometric based multi-server authentication scheme for social multimedia networks," *Multimedia Tools Appl.*, vol. 75, no. 20, pp. 12705–12725, 2016.

[10] Y. S. Lee, E. Kim, S. J. Seok, and M. S. Jung, "A smart card-based user authentication scheme to ensure the PFS in multi-server environments," *IEICE Trans. Commun.*, vol. E95–B, no. 2, pp. 619–622, 2012.

[11] D. B. He, "Security flaws in a biometrics-based multi-server authentication with key agreement scheme," Tech. Rep. 2011/365, ePrint Archive.

[12] D. He and H. Hu, "Cryptanalysis of a smart card-based user authentication scheme for multi-server environments," *IEICE Trans. Commun.*, vol. E95–B, no. 9, pp. 3052–3054, 2012.

[13] X. Cao and S. Zhong, "Breaking a remote user authentication scheme for multi-server architecture," *IEEE Commun. Lett.*, vol. 10, no. 8, pp. 580–581, Aug. 2006.

[14] W. S. Juang, "Efficient multi-server password authenticated key agreement using smart cards," *IEEE Trans. Consum. Electron.*, vol. 50, no. 1, pp. 251–255, Feb. 2004.

[15] C. C. Chang and J. S. Lee, "An efficient and secure multi-server password authentication scheme using smart cards," in *Proc. 3rd Int. Conf. Cyberworlds*, Nov. 2004, pp. 417–422.

[16] W. J. Tsaur, C. C. Wu, and W. B. Lee, "A smart card-based remote scheme for password authentication in multi-server Internet services," *Comput. Standards Interfaces*, vol. 27, no. 1, pp. 39–51, 2004.

[17] W. J. Tsaur, C. C. Wu, and W. B. Lee, "An enhanced user authentication scheme for multi-server Internet services," *Appl. Math. Comput.*, vol. 170, no. 1, pp. 258–266, 2005.

[18] J. L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Comput. Security*, vol. 27, no. 3–4, pp. 115–121, 2008.

[19] J. S. Chou, Y. Chen, C. H. Huang, and Y. S. Huang, "Comments on four multi-server authentication protocols using smart card," IACR Cryptol. ePrint Archive, 2012/406, 2012. [Online]. Available: https://eprint.iacr.org/2012/406.pdf

[20] D. B. He and H. Hu, "Cryptanalysis of a smart card-based user authentication scheme for multi-server environments," *IEICE Trans. Commun.*, vol. E95–B, no. 9, pp. 3052–3054, 2012.

[21] L. Gong, R. Needham, and R. Yahalom, "Reasoning about belief in cryptographic protocols," in *Proc. of IEEE Comput. Soc. Symp. Res. Security Privacy*, Oakland, CA, USA, May 1990, pp. 234–248.

[22] H. C. Hsiang and W. K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," *Comput. Standards Interfaces*, vol. 31, no. 6, pp. 1118–1123, 2009.

[23] S. K. Sood, A. K. Sarje, and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," *J. Netw. Comput. Appl.*, vol. 34, no. 2, pp. 609–618, 2011.

[24] C. C. Lee, T. H. Lin, and R. X. Chang, "A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards," *Expert Syst. App.*, vol. 38, no. 11, pp. 13863–13870, 2011.

[25] X. Li, Y. P. Xiong, J. Ma, and W. D. Wang, "An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards," *J. Netw. Comput. App.*, vol. 35, no. 2, pp. 763–769, 2012.

[26] A. T. B. Jin, D. N. C. Ling, and A. Goh, "Bio-hashing: Two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recog.*, vol. 37, no. 11, pp. 2245–2255, 2004.

[27] A. Lumini and N Loris, "An improved Bio-hashing for human authentication," *Pattern Recog.* vol. 40, no. 3, pp. 1057–1065, 2007.

[28] Y. P. Liao and S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," *Comput. Standard Interfaces*, vol. 31, no. 1, pp. 24–29, 2009.

[29] Y. S. Lee, E. Kim, S. J. Seok, and M. S. Jung, "A smart card-based user authentication scheme to ensure the PFS in multi-server environments," *IEICE Trans. Commun.*, vol. E95–B, no. 2, pp. 619–622, 2012.

[30] D. Mishra, A. Das, and S. Mukhopadhyay, "A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards," *Expert Syst. App.*, vol. 41, no. 18, pp. 8129–8143, 2014.

Authorized licensed use limited to: Abu Dhabi University. Downloaded on March 13,2025 at 11:15:12 UTC from IEEE Xplore. Restrictions apply.

[31] C.-T. Chen and C.-C. Lee, "A two-factor authentication scheme with anonymity for multiserver environments," *Security Commun. Netw.*, vol. 8, no. 8, pp. 1608–1625, 2015.
[32] C.-C. Chang, T.-F. Cheng, and W.-Y. Hsueh, "A robust and efficient dynamic identity-based multi-server authentication scheme using smart cards," *Int. J. Commun. Syst.*, vol. 29, pp. 290–306, 2014.
[33] M. Chuang and M. Chen, "An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," *Expert Syst. Appl.*, vol. 41, pp. 1411–1418, 2014.
[34] Y. Lu, L. Li, H. Peng, and Y. Yang, "A biometrics and smart cards-based authentication scheme for multi-server environments," *Security Commun. Netw.*, vol. 8, no. 17, pp. 3219–3228, 2015.
[35] R. S. Pippal, C. D. Jaidhar, and S. Tapaswi, "Robust smart card authentication scheme for multi-server architecture," *Wireless Personal Commun.*, vol. 72, no. 1, pp. 729–745, 2013.
[36] H. Lin, F. Wen, and C. Du, "An improved anonymous multi-server authenticated key agreement scheme using smart cards and biometrics," *Wireless Personal Commun.*, vol. 84, no. 4, pp. 2351–2362, 2015.
[37] K.-H. Yeh, "A provably secure multi-server based authentication scheme," *Wireless PersOnal Commun.*, vol. 79, no. 3, pp. 1621–1634, 2014.
[38] D. Mishra, "Design and analysis of a provably secure multi-server authentication scheme," *WireLess PersOnal Commun.*, vol. 86, no. 3, pp. 1095–1119, 2016.
[39] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proc. ACM Conf. Comput. Comm. Security*, Fairfax, VA, USA, 1993, pp. 62–73.
[40] X. Li *et al.*, "A novel chaotic maps-based user authentication and key agreement protocol for multi-server environments with provable security," *Wireless Personal Commun.*, vol. 89, no. 2, pp. 569–597, 2016.
[41] H. Shen, C. Gao, D. He, and L. Wu, "New biometrics-based authentication scheme for multi-server environment in critical systems," *J. Ambient Intel. Humanized Comput.*, vol. 6, no. 6, pp. 825–834, 2015.
[42] H. Zhu, "Flexible and password-authenticated key agreement scheme based on chaotic maps for multiple servers to server architecture," *Wireless Personal Commun.*, vol. 82, no. 3, pp. 1697–1718, 2015.
[43] J. L. Tsai and N. W. Lo, "A chaotic map-based anonymous multi-server authenticated key agreement protocol using smart card," *Int. J. Commun. Syst.*, vol. 28, no. 13, pp. 1955–1963, 2015.
[44] P. Jiang, Q. Wen, W. Li, Z. Jin, and H. Zhang, "An anonymous and efficient remote biometrics user authentication scheme in a multi server environment," *Frontiers Comput. Sci.*, vol. 9, no. 1, pp. 142–156, 2015.
[45] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, and N. Kumar, "An efficient and practical smart card based anonymity preserving user authentication scheme for TMIS using elliptic curve cryptography," *J. Med. Syst.*, vol. 39, no. 11, pp. 1–18, 2015.
[46] J. L. Tsai and N. W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Syst. J.*, vol. 9, no. 3, pp. 805–815, Sep. 2015.
[47] D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen, "Efficient privacy-aware authentication scheme for mobile cloud computing services," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1621–1631, Jun. 2016.
[48] A. Irshad, M. Sher, H. F. Ahmad, B. A. Alzahrani, S. A. Chaudhry, and R. Kumar, "An improved Multi-server Authentication Scheme for Distributed Mobile Cloud Computing Services," *KSII Trans. Internet Inf. Sys.*, vol. 10, no. 12, pp. 5529–5552, 2016.
[49] R. Amin, S. K. Islam, G. P. Biswas, D. Giri, M. K. Khan, and N. Kumar, "A more secure and privacy-aware anonymous user authentication scheme for distributed mobile cloud computing environments," *Security Commun. Netw.*, vol. 9, no. 17, pp. 4650–4666, 2016.
[50] Shamus Software Ltd., Miracl library, 2003. [Online]. Available: http://www.shamus.ie/index.php?page=home

Azeem Irshad received the Master's degree from Arid Agriculture University, Rawalpindi, Pakistan. He is currently working toward the Ph.D. degree in security for multiserver architectures from International Islamic University, Islamabad, Pakistan.

He has authored or coauthored more than 40 research publications in international journals and proceedings. His research interests include strengthening of authenticated key agreements in SIP multimedia, IoT, WBAN, TMIS, WSN, Ad hoc Networks, e-health clouds, and multiserver architectures.

Shehzad Ashraf Chaudhry received the Master's degree (with distinction) and the Ph.D. degree from International Islamic University Islamabad, Pakistan, in 2009 and 2016, respectively. He was awarded Gold Medal for achieving 4.0/4.0 CGPA in his Masters.

He is currently an Assistant Professor with the Department of Computer Science and Software Engineering, International Islamic University. He has authored more than 62 scientific publications, which appeared in different international journals and proceedings, including 42 in SCI/E journal. His research interests include lightweight cryptography, elliptic/hyper elliptic curve cryptography, multimedia security, e-payment systems, MANETs, SIP authentication, smart grid security, IP multimedia subsystem, and next-generation networks.

Muhammad Sher received the Ph.D. degree in computer science (NGN security) from Technical University, Berlin, Germany, in 2007.

He is the Chairman of the Department of Computer Science and Software Engineering, International Islamic University, Islamabad, Pakistan. He has authored or co-authored more than 125 scientific research publications. His research interests include NGN, SIP, and Multimedia security.

Bander A. Alzahrani received the Ph.D. degree in computer science from University of Essex, U.K., in 2015.

He is an Assistant Professor at King Abdulaziz University, Saudi Arabia. He has published more than 18 research papers in International Journals and conferences. His research interests include Network security, Information centric networks.

Saru Kumari received the Ph.D. degree in mathematics in 2012 from Chaudhary Charan Singh (CCS) University, Meerut, India.

She is currently an Assistant Professor with the Department of Mathematics, CCS University, Meerut, India. Her research interests include information security and security of WSN.

Xiong Li received the master's degree in mathematics and cryptography from Shaanxi Normal University, Xi'an, China, in 2009 and the Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications, Beijing, China, in 2012.

He is currently a Lecturer at Hunan University of Science and Technology, Xiangtan, China.

Fan Wu received the master's degree in computer science and theory from Xiamen University, Xiamen, China, in 2008.

He is currently working in Xiamen Institute of Technology. His current research interests include security in wireless communication networks.

Authorized licensed use limited to: Abu Dhabi University. Downloaded on March 13,2025 at 11:15:12 UTC from IEEE Xplore. Restrictions apply.