Wireless Personal Communications (2020) 111:991–1017
https://doi.org/10.1007/s11277-019-06897-8

Improved Biometric-Based Mutual Authentication and Key Agreement Scheme Using ECC

Shreeya Swagatika Sahoo[1] · Sujata Mohanty[1] · Banshidhar Majhi[1]

Published online: 25 October 2019

Abstract
Recently, biometric based authentication scheme gains popularity due to its high security, integrity, and authenticity properties. In the recent past, Qi et al. improved Chaudhry et al.'s scheme, which is susceptible to the DoS attack and fails to achieve perfect forward secrecy. In this paper, we analyze Qi et al.'s biometric based authentication scheme and show that the scheme cannot withstand key compromise impersonation attack, offline password guessing attack, and known session-specific temporary information attack. We proposed an improved biometric based authentication scheme using Elliptic Curve Cryptography (ECC) with more security functionalities. Further, we prove the mutual authentication and session key security of the proposed scheme using Burrows–Abadi–Needham (BAN) logic and Random Oracle Model (ROM). Moreover, the security analysis endorses that the proposed scheme is robust enough to provide protection against all well known attacks. The simulation results using the AVISPA tool show that the proposed scheme is secure and achieves its goal.

Keywords Authentication · Biometric · Smart card · BAN Logic · Random Oracle Model · AVISPA

1 Introduction

In recent time, the advancement of the Internet and telecommunication technologies provide various online services, such as banking, telecommuting, gaming, e-health, etc. Though these various services make everyday life easy and convenient; the user accesses these services through an insecure channel making it an easy target for the adversary. Thus, to protect the sensitive information from the adversary, authentication among the participants is needed. To ensure the authenticity of the user and server, mutual authentication and session key security plays a vital role. Authentication can be achieved using single factor (password), two factor (smart card), and three-factor (biometric). Nevertheless, the password may be forgotten, and smart card may be shared, lost, or stolen. The biometric-based schemes have no such issues, and further, it is very difficult to copy, forge, and guess.

✉ Shreeya Swagatika Sahoo
shreeya.swagatika@gmail.com

[1] National Institute of Technology, Rourkela, India

So, the biometric-based authentication schemes attracted wide attention of researchers. To design a secure authentication scheme, cryptographic functions such as RSA cryptosystem, ECC cryptosystem, bilinear pairing, one-way hash function, etc. are used. To ensure the requirement of practical applications, many password, smart card, and biometric-based schemes have been proposed using several cryptographic functions [1–4].

In 1981, Lamport proposed a password-based authentication protocol using one-way hash functions which store the hash value of the password in the server's database [5]. Later, to provide security and efficiency, several password based remote user authentication schemes have been proposed for various applications [6–9]. Later, it was shown that password-based authentication schemes could be easily breached if the database is compromised or revealed. To overcome these weaknesses, two factor based authentication schemes have been suggested [10–20]. However, two factor schemes have some weaknesses such as the smart card can be lost, shared with others, or the information can be extracted from it. Thus, biometric-based authentication schemes have been suggested based on different cryptosystem [21–26]. Although, both RSA and ECC facilitate the same level of security, ECC cryptosystem is more efficient due to its less key length size.

In 2013, Yoon et al. [27] suggested a biometric-based remote user authentication scheme using ECC for multi-server environment. Yeh et al. [28] suggested a biometric-based authentication scheme for client-server networks. However, Wu et al. [29] found that Yeh et al.'s scheme could not resist impersonation attack and failed to achieve session key agreement, mutual authentication. Later, Kim et al. [30] pointed out that Yoon et al.'s scheme is not secure against offline password guessing attack, lost smart card attack. He et al. [31] also found that Yoon et al.'s scheme is susceptible to insider attack and impersonation attack. However, He et al. pointed out that both Yoon et al. and Kim et al.'s scheme suffer from the impersonation attack. Further, Odelu et al. [32] proved that He et al.'s scheme is insecure against the replay attack, impersonation attack, and known session specific information attack.

Based on analyzing Tan et al.'s [33] scheme, Arshad et al. [34] suggested an improved biometric-based authentication scheme using ECC. Afterward, Lu et al. [35] observed that Tan et al.'s scheme could not resist user impersonation attack, off-line password guessing attack, and suggested an enhanced authentication scheme. Nevertheless, Chaudhry et al. [36] pointed out that Lu et al.'s scheme is vulnerable to user impersonation attack, server impersonation attack and fail to achieve user anonymity, user traceability. In 2015, Mir et al. [37] presented an ECC based authentication scheme for telemedicine networks. Furthermore, Chaudhry et al. [38] proved that Mir et al.'s scheme suffers from lost smart card attack and could not achieve user anonymity. However, Qi et al. [39] found that Chaudhry et al.'s scheme failed to provide perfect forward secrecy and could not withstand denial of service attack. Then, Qi et al. suggested an improved new scheme claiming that their scheme can resist various attack. However, in this paper, we point that Qi et al.'s scheme cannot prevent the known session-specific temporary information attack, key compromise impersonation attack, and offline password guessing attack.

We present a biometric-based authentication scheme using ECC. The contributions of the proposed scheme are outlined as follows.

1. We analyzed the security of Qi et al.'s scheme and demonstrated that the scheme is insecure against key compromise impersonation attack, offline password guessing attack, and known session-specific temporary information attack. To overcome the above weaknesses, we present a biometric-based authentication scheme using ECC.

2. The formal proof has been done with the help of ROM which proves the session key security of the scheme and is secured against an adversary for retrieving user's identity and secret key.
3. The mutual authentication of the proposed scheme has done using widely accepted BAN logic. Moreover, informal security analysis shows that the scheme is secure and can withstand several known attacks.
4. Further, we simulated our scheme using the AVISPA tool, which shows that the scheme is secure under OFMC and CL-AtSe backends.
5. The proposed scheme provides high security along with several security features and less communicational cost compared to other existing schemes.

The remaining part of this paper is organized as follows: Sect. 2 describes the mathematical preliminaries such as hash function, ECC, and Bio-hashing. We briefly review Qi et al.'s scheme and point out the weaknesses of their scheme in Sect. 3 and Sect. 4 respectively. In Sect. 5, we proposed a biometric-based authentication scheme using ECC. Formal and informal security analysis of the proposed scheme are demonstrated in Sect. 6. The simulation and the performance analysis of the scheme are presented in Sect. 7 and Sect. 8 respectively. Finally, Sect. 9 presents the conclusion.

2 Mathematical Preliminaries

This section discusses the mathematical preliminaries used for the proposed scheme.

2.1 Hash Function

The properties of one-way hash function $h : \{0,1\}^* \rightarrow \{0,1\}^l$ is to takes an arbitrary length of input string $k \epsilon \{0,1\}^*$, and generates fixed length l of output string. It is considered as a secure hash function which has the following properties:

1. For a given hash value y, it is difficult to find any input k such that $y = h(k)$.
2. To compute k_2 for a given k_1 is computationally infeasible, such that $k_1 \neq k_2$, where $h(k_1) = h(k_2)$.
3. It is difficult to find two different message (k_1, k_2) such that $h(k_1) = h(k_2)$.

Definition 1 (*Collision-Resistant One-way Hash Function*) The hash function $h : \{0,1\}^* \rightarrow \{0,1\}^l$ is considered as a deterministic algorithm which takes an arbitrary length of binary string and produces l length of the output string. If $ADV_A^{HASH}(t)$ is a A's advantage in finding a collision, then we have

$$ADV_A^{HASH}(t) = \Pr[(k1, k2) \ \varepsilon \ _R A \ : \ k1 \neq k2, h(k1) = h(k2)]$$

$\Pr[S]$ denotes the probability of random event S and $(k1, k2) \ \varepsilon \ _R A$ denotes the pair $(k1, k2)$ randomly selected by an adversary A_k. An adversary computes probability in advantage over the random value with execution time t. if $Adv_A^{HASH}(t) \leqslant \varepsilon$, for any sufficiently small $\varepsilon \geqslant 0$, then hash fuction $h(.)$ is called collision-resistant.

2.2 Elliptic Curve Cryptography (ECC)

The ECC equation is $y^2 = x^3 + ax + b$ over the finite field Z_p where a, $b \in Z_p$ and a non-singular elliptic curve must satisfy $4a^3 + 27b^2\ mod\ p \neq 0$. In ECC, the scalar multiplication is defined as the repeated addition. Let G be a base point on elliptic curve E_p whose order be n. If $G \in E_p$, then $nG = G + G + \cdots n\ (n\ times)$.

Definition 2 (*Elliptic Curve Discrete Logarithm Problem* (*ECDLP*)) Against two arbitrary points $G, T\ \varepsilon\ Z_p(a, b)$, computes a scalar n such that $T = nG$. An adversary can compute n during the polynomial time t is $ADV_A^{ECDLP}(t) = Prb[(A(G, T)) = x : x\varepsilon Z_p]$. The ECDLP infers that $ADV_A^{ECDLP}(t) \leqslant \varepsilon$.

2.3 Bio-hashing

The biometric technology plays an important role in the authentication system to validate a legal user. Generally, hash functions produce huge differences in hash value because of the minute change in inputs. The biometric characteristics such as the face, fingerprint, palmprint etc. may behave differently each time these are collected. However, a little deviation of biometric data or a change in the order of data input will result in a huge difference in hash values. To overcome this drawback, bio-hashing are used in which a legal user can be authenticated in case the user's biometric data has a little deviation.

2.4 Adversarial Model

Here, we consider the following capabilities of an adversary. The assumption is an adversary can extract the secure information from the smart card using power analysis attacks or reverse engineering procedures [40, 41]. The Dolev–Yao [42] threat model has been used in which both user and server communicate each other over an insecure channel. An attacker may eavesdrop, modify and replay the messages over an insecure channel.

3 Review of Qi et al.'s Scheme

This section presents a brief review of Qi et al.'s [39] authentication scheme. Qi et al.'s scheme has four phases, namely system initialization phase, user registration phase, login and authentication phase, password change phase. There are two participants, namely user (U_m) and server (S). The phases are demonstrated in Table 2 and Table 3 respectively. The notations used in Qi et al.'s scheme are listed in Table 1.

Table 1 Notation used

Notation	Description
U_m	mth User
S	Server
SC	Smart card
ID_m	User's (mth) identity
PW_m	User's (mth) password
B_m	User's biometric
x, y	Server's long term private key
$h(\cdot)$	Secure one way hash function
$E_{(k)}/D_{(k)}$	Encryption/decryption operations
$\Vert$	Concatenation operation
$\oplus$	Bitwise exclusive OR operator
KS	Session key of user and server
A_k	Adversary

Table 2 Registration phase of the Qi et al.'s scheme

User (U_m)		Server(S)
Chooses ID_m, PW_m and biometric B_m		
Computes $MB_m = h(PW_m \Vert H(B_m))$		
	$\dashrightarrow \{ID_m, MB_m\}$	
		Generates a random number y
		$W_m = y \oplus h(ID_m \Vert x)$
		$V_m = y \oplus MB_m$
		$A_m = h(ID_m \Vert MB_m)$
	$\dashleftarrow \{A_m, V_m, W_m, h(.)\}$	
	Secure channel $---\rightarrow$	
	Insecure channel$\longrightarrow$	

3.1 Initialization Phase

The server selects a large distinct prime number p over a finite field Z_p on an elliptic curve. Server chooses a secure one way hash function $h : \{0, 1\} \rightarrow Z_p^*$ and a bio-hashing operator $H : \{0, 1\} \rightarrow Z_p^*$. The server generates its private key $x \epsilon Z_n^*$ and computes the public key $P_{pub} = x.G$, where G is the base point.

Table 3 Login and authentication phase of Qi et al.'s scheme

User (U_m)		Server(S)
The user puts his smart card and enters		
ID'_m, PW'_m, and B_m		
$MB'_m = h(PW_m \Vert H(B_m))$		
$A_m \overset{?}{=} h(ID_m \Vert MB'_m)$		
Generate r and calculates		
$y' = V_m \oplus (MB'_m)$		
$S_1 = r.G, S_2 = r.P_{pub}$		
$C_m = E_k(ID_m \Vert h(PW_m \Vert r) \Vert W_m)$		
$Auth_i = h(y \Vert ID_m \Vert h(PW_m \Vert r) \Vert W_m \Vert S_1 \Vert T_i)$		
	$\xrightarrow{\{Auth_i, C_m, S_1, T_i\}}$	
		Verify $(T'_i - T_i) \leq \triangle T$
		Computes $S'_2 = x.S_1$
		$(ID'_m \Vert h(PW_m \Vert r) \Vert W'_m) = D_k(C_m)$
		$y' = W'_m \oplus h(ID'_m \Vert x)$
		$Auth'_i = h(y' \Vert ID'_m \Vert h(PW_m \Vert r) \Vert W'_m \Vert S_1 \Vert T_i)$
		If $(Auth_i \overset{?}{=} Auth'_i)$, aborts if not true
		Generates a random number t
		$S_3 = t.G, S_4 = t.S_1$
		$KS = h(S_4 \Vert h'(PW_m \Vert r))$
		$Auth_s = h(y' \Vert KS \Vert S_1 \Vert S_3 \Vert T_s)$
	$\xleftarrow{\{Auth_s, S_3, T_s\}}$	
Verify $(T'_s - T_s) \leq \triangle T$		
Computes $S'_4 = r.S_3$		
$KS' = h(S'_4 \Vert h(PW_m \Vert r))$		
$Auth'_s = h(y' \Vert KS' \Vert S_1 \Vert S_3 \Vert T_s)$		
Checks $(Auth'_s \overset{?}{=} Auth_s)$		
If not true, then ABORT		
$Auth_{is} = h(h(PW_m \Vert r) \Vert KS' \Vert S_3)$		
	$\xrightarrow{\{Auth_{is}\}}$	
		$Auth_{is} = h(h'(PW_m \Vert r) \Vert KS \Vert S_3)$
		If $(Auth_{is} = Auth_{si})$, then
		Session key is verified
		Otherwise, Abort the session.

3.2 Registration Phase

To get service from the server, the user first registers himself to the server by performing the following steps.

Step 1 U_m chooses his identity ID_m, password PW_m and imprints his biometric B_m via a sensor. Then, computes $MB_m = h(PW_m \oplus H(B_m))$. Now, U_m sends $\{ID_m, PW_m, H(B_m)\}$ to the server through a secure channel.

Step 2 After receiving the message, server chooses a secret key x and computes $W_m = y \oplus h(ID_m \parallel x), V_m = y \oplus MB_m, A_m = h(ID_m \| MB_m)$, where y is a random number generated by server. Then, S stores $\{W_m, V_m, A_m, h(.)\}$ into the SC and issues it to U_m securely.

3.3 Login and Authentication Phase

To login into the system, U_m inserts his smart card into the card reader to access the server. The following steps are carried out during the login phase.

Step 1 User enters his ID_m, PW_m and imprints biometric B_m through sensor.

Step 2 SC computes $MB'_m = h(PW_m \| H(B_m))$ and checks whether $A_m \overset{?}{=} h(ID_m \| MB'_m)$ or not. If both are equal, then the smart card generates a random number r and compute $y' = V_m \oplus (MB'_m), S_1 = r.G$, $S_2 = r.P_{pub}, C_m = E_k(ID_m \| h(PW_m \| r) \| W_m), Auth_i = h(y \| ID_m \| h(PW_m \| r) \| W_m \| S_1 \| T_i)$. Then, sends the login message $\{Auth_i, C_m, S_1, T_i\}$ to the server through an open channel.

Step 3 After obtaining the login message, S first check the time stamp $(T'_i - T_i) \leq \triangle T$. After successful time-stamp verification, S computes $S'_2 = x.S_1$, $D_{S'_2}(C_m) = (ID'_m \| h(PW_m \| r) \| W'_m)$, $y' = W'_m \oplus h(ID'_m \| x)$, $Auth'_i = h(y' \| ID'_m \| h(PW_m \| r) \| W'_m \| S_1 \| t_i)$.

Step 4 If $(Auth_i \overset{?}{=} Auth'_i)$ satisfies, then the server generates a random number t and calculates $S_3 = t.G, S_4 = t.S_1, KS = h(S_4 \| h'(PW_m \| r)), Auth_s = h(y' \| KS \| S_1 \| S_3 \| T_s)$. Then, sends $\{Auth_s, S_3, T_s\}$ to the user through a public channel.

Step 5 U_m first checks the time stamp $(T'_s - T_s) \leq \triangle T$. After successful time-stamp verification, U_m computes $S'_4 = r.S_3$, $KS' = h(S'_4 \| h(PW_m \| r))$, $Auth_s = h(y' \| KS' \| S_1 \| S_3 \| T_s)$. Then, he verifies whether $(Auth'_s \overset{?}{=} Auth_s)$.

Step 6 If it is true, U_m calculates $Auth_{is} = h(h(PW_m \| r) \| KS' \| S_3)$ and sends $Auth_{is}$ to S through a public channel.

Step 7 Upon receiving the message from user, S computes $Auth_{si} = h(h'(PW_m \| r) \| KS \| S_3)$ and compares with $Auth_{is}$. If both are same, then both U_m and S are mutually authenticated and agrees to communicate through the shared session key.

3.4 Password Change Phase

The user updates his password without interacting with the server as follows.

Step 1 User enters his identity ID_m, password PW_m, and scans his biometric B_m. The smart card computes $MB'_m = h(PW_m \| H(B_m))$. Then, verifies the condition $A_m = h(ID_m \| MB'_m)$.

Step 2 If both are equal, then the smart card asks the user for a new password PW_m^{new}. After entering the new password, SC computes $MB_m^{new} = h(PW_m^{new} \| H(B_m))$, $V_m^{new} = V_m \oplus MB'_m \oplus MB_m^{new}$, $A_m^{new} = h(ID_m \| MB_m^{new})$ and replaces V_m, A_m with V_m^{new}, A_m^{new} respectively.

4 Cryptanalysis of Qi et al.'s Scheme

In this section, we have demonstrated that Qi et al.'s scheme is susceptible to key compromise impersonation attack, offline-password guessing attack, and known session specific temporary information attack.

4.1 Key Compromise Impersonation Attack

Key Compromise Impersonation (KCI) attack is a popular attack, in which the private key of the participating entity is revealed. Qi et al.'s scheme could not withstand the KCI attack. Suppose, the private key x is revealed. An adversary can perform KCI attack as per the following steps.

Step 1 Let an adversary can eavesdrop the login message $\{Auth_i, C_m, S_1, T_i\}$ from the public channel. Then, verifies the time stamp and computes $S_2^* = x.S_1, D_k'(C_m) = (ID_m \| h'(PW_m \| r) \| W_m'), y' = W_m' \oplus h(ID_m \| x)$.
Step 2 Now, he successfully validates and generates his own random number t^*. Computes $S_3 = t^*.G, S_4 = t^*.S_1, KS^* = h(S_4^* \| h'(PW_m \| r)), Auth_i^* = h(y' \| KS^* \| S_1 \| S_3 \| T_s)$ and sends it to the user.
Step 3 Then, user verify $Auth_s = Auth_s'$ and this verification will get true. In this manner, an adversary may lunch a successful impersonation attack and fool the user.

4.2 Offline Password Guessing Attack

From the aforementioned analysis, the adversary can obtain $h(PW_m \| r)$ by decrypting C_m. He chooses a new password PW_a and computes $Auth_i = h(y' \| ID_m' \| h(PW_a \| r) \| W_m' \| S_1 \| T_i)$ where ID_m', y',and W_m' known to the attacker. Again S_1 and T_i eavesdrop from the public channel. The check continues until the correct password is obtained. Thus, the scheme could not resist offline password guessing attack.

4.3 Known Session-Specific Temporary Information Attack

Let, an adversary get the user's session random number r unexpectedly. Then, Qi et al.'s scheme has the following drawback:

Step 1 Both user and server compute the session key KS as $KS = h(S_4 \| h(PW_m \| r)) = h(S_4' \| h(PW_m \| r)) = h(t.r.S_3 \| h(PW_m \| r))$. An adversary can calculates the session key using known session random number r.
Step 2 An adversary intercept the login message $\{Auth_i, C_m, S_1, T_i\}$ sent to the server and checks whether $r.G$ matches with S_1. If it matches, adversary confirms that r corresponds to the login message. The adversary sends the login message to the server without any modification. Upon receiving the message, the server will check the validity and respond the message $\{Auth_s, S_3, T_s\}$. As the adversary knows the r, so he can easily compute $S_4' = r.S_3$. As discussed in Sect. 4.1 an adversary can get $h(PW_m \| r)$, now the adversary can easily compute the session key as $KS = h(S_4' \| h(PW_m \| r))$ and compute $Auth_{is} = h(h(PW_m \| r) \| KS' \| S_3)$ without knowledge of a valid user. Thus, this scheme could not achieve session key security.

5 Proposed Scheme

To overcome the flaws of the Qi et al.'s scheme, we proposed an improved three-factor based authentication scheme. The proposed scheme consists of five phases: initialization phase, registration phase, login phase, authentication phase, and password change phase. There are two participants, namely server (S), and user (U_m). We used the same notations as presented in Table 1. The details of each phase are illustrated below.

5.1 Initialization Phase

The S chooses a large distinct prime number p over a finite field Z_p on an elliptic curve. A non-singular elliptic curve equation is defined as $y^2 = x^3 + ax + b$, where $a, b \epsilon Z_p$ and must satisfy $4a^3 + 27b^2 \ mod \ p \neq 0$.

The server selects a point P on the curve $E_p(a, b)$ over a finite field Z_p. Then, chooses a secret key y and computes $Pub = y \times P$. Now, S declares $\{x, y\}$ as its private key and $\{E, P, Pub\}$ as public key.

5.2 Registration Phase

A new user needs to register with the server by performing the following steps.

Step 1 The user (U_m) freely selects his identity ID_m, password PW_m, and personal biometric B_m at the sensor.

Table 4 Registration phase of the proposed scheme

User (U_m)		Server(S)
Choose ID_m, PW_m and biometric B_m.		
Computes $PW_{nw} = h(PW_m \| H(B_m))$		
	- - → $\{ID_m, PW_{nw}, H(B_m)\}$	
		Generate private key x and y, and pseudonym identity PID_m $M_i = h(ID_m \| H(B_m).P = (P_x, P_y)$ $N_m = h(ID_m \| x)$ $P_m = N_m \oplus h(ID_m \| PW_{nw})$ $Q_m = h(ID_m \| PW_{nw} \| N_m)$ $UID_m = ID_m \oplus h(x \| y)$
	← - - $\{P_m, Q_m, UID_m, PID_m, h(.)\}$	
		Stores $\{ID_m, PID_m, M_i\}$
	Secure channel − − − →	
	Insecure channel———→	

Step 2 Then, U_m computes his dynamic password $PW_{nw} = h\{(PW_m \| H(B_m))\}$ and sends the message $\{ID_m, PW_{nw}, H(B_m)\}$ to the server S.
Step 3 Upon receiving registration message $\{ID_m, PW_{nw}, H(B_m)\}$, the S records $H(B_m)$ for future use. Then, selects a private key x and calculates $M_i = h(ID_m \| H(B_m).P = (P_x, P_y)$, $N_m = h(ID_m \| x)$, $P_m = N_m \oplus h(ID_m \| PW_{nw})$, $Q_m = (ID_m \| PW_{nw} \| N_m)$, $UID_m = ID_m \oplus h(x \| y)$.
Step 4 S generate a pseudonym identity PID_m for the user U_m. For new user registration, the server sets N = 0, otherwise, N = N+1 where N is the number maintained by the server.
Step 5 Finally, S embedded the parameters $\{UID_m, PID_m, P_m, Q_m, h(.)\}$ into the smart card and issues it to the user. Server stores $\{UID_m, ID_m, B_m\}$ for future use. The details of the registration phase are described in Table 4.

5.3 Login Phase

In order to login to the server S, the U_m performs the following steps:

Step 1 The user puts his smart card into the card reader and imprints his biometric B_m on the device. Also, inputs his user name (ID_m) along with the password (PW_m). SC computes $PW_{nw}^* = h(PW_m \| H(B_m))$, $N_m^* = P_m \oplus h(ID_m \| PW_{nw}^*)$, $Q_m^* = (ID_m \| PW_{nw}^* \| N_m^*)$.
Step 2 Then, SC compares computed Q_m^* with the received parameter Q_m. If the match goes wrong, then the smart card aborts the session. Otherwise, it continues for the next step.
Step 3 The SC generates a random number n_1 and computes $M_i^* = h(ID_m \| H(B_m))$. $P = (P_x, P_y), Z_1 = n_1.P$, $Z_2 = n_1.Pub$, $Z_3 = E_{(P_x)}(ID_m \| Z_1 \| n_1), Z_3 = n_1 \oplus M_i^*, Z_4 = h(ID_m \| M_i^* \| N_m^* \| n_1 \| Z_1)$. Finally, U_m sends login message $\{UID_m, PID_m, Z_3, Z_4\}$ to the server through a public channel.

5.4 Authentication Phase

After getting the login request, $\{UID_m, PID_m, Z_3, Z_4\}$, both S and U_m perform the following steps. Table 5 represents the details of the authentication phase.

Step 1 The S calculates $ID_m = UID_m \oplus h(x \| y)$ and searches for ID_m from the data base. If exists, then computes $n_1^* = Z_3 \oplus M_i$, $Z_1^* = n_1^*.P$, $Z_4^* = h(ID_m \| M_i^* \| N_m \| n_1^* \| Z_1)$. Now, checks if $Z_4 \overset{?}{=} Z_4^*$ or not. If it does not hold then, server terminates the session. Otherwise, proceeds to the next steps.
Step 2 S generates a random number n_2 and computes $Y_1 = n_2.P$, $Y_2 = E_{p(x)}(n_2 \| Y_1)$, $S_1 = n_2.Z_1^*$, $KS = h(S_1 \| N_m)$, $Auth_s = h(SID_j \| KS \| N_m \| Y_1)$. Then, sends $\{Y_2, Auth_s\}$ to the U_m.
Step 3 After receiving the authentication message from the S, user decrypts the message $D_{p(x)}(Y_2) = (n_2 \| Y_1)$ and computes $T_1 = n_1 \cdot Y_1, KS = h(T_1 \| N_m^*), Auth_s^* = h(SID_j \| KS \| N_m^* \| Y_1)$.

Table 5 Login and authentication agreement phase

User(U_m)		Server(S)
User inserts SC and		
Inputs ID_m, PW_m, B_m		
SC calculates		
$PW^*_{nw} = h(PW_m \| H(B_m))$		
$N^*_m = P_m \oplus h(ID_m \| PW^*_{nw})$		
$Q^*_m = h(ID_m \| PW^*_{nw} \| N^*_m)$		
Checks $Q_m \overset{?}{=} Q^*_m$		
If true, generate a random number n_1		
$M^*_i = h(ID_m \| H(B_m)).P = (P_x, P_y)$		
$Z_1 = n_1.P, Z_2 = n_1.Pub$		
$Z_3 = n_1 \oplus M^*_i$		
$Z_4 = h(ID_m \| M^*_i \| N^*_m \| n_1 \| Z_1)$		
	$\xrightarrow{\{UID_m, PID_m, Z_3, Z_4\}}$	
		Computes $ID_m = UID_m \oplus h(x \| y)$
		Search for M_i, if ID_m exist
		Computes $n^*_1 = Z_3 \oplus M_i$
		$Z^*_1 = n^*_1.P$
		$Z^*_4 = h(ID_m \| M_i \| N^*_m \| n^*_1 \| Z^*_1)$
		Checks $Z_4 \overset{?}{=} Z^*_4$
		If true, then user is valid
		Server generates n_2 and computes
		$Y_1 = n_2.P, Y_2 = E_{p(x)}(n_2 \| Y_1)$
		$S_1 = n_2.Z^*_1, KS = h(S_1 \| N_m)$
		$Auth_s = h(SID_j \| KS \| N_m \| Y_1)$
	$\xleftarrow{\{Y_2, Auth_s\}}$	
Computes $D_{p(x)}(Y_2) = (n_2 \| Y_1)$		
$T_1 = n_1.Y_1, KS = h(T_1 \| N^*_m)$		
$Auth^*_s = h(SID_j \| KS \| N^*_m \| Y_1)$		
Verifies $Auth_s \overset{?}{=} Auth^*_s$		
If the condition fails, session is terminated		
Otherwise, the user calculates M_3		
$Auth_{is} = h(ID_m \| KS \| n_1 \| n_2)$		
	$\xrightarrow{\{Auth_{is}\}}$	
		After getting the message, server will verify
		If $Auth_{is} \overset{?}{=} h(ID_m \| KS \| n_1 \| n_2))$,
		Then session key is verified, otherwise
		terminate the session

Step 4 Now, U_m verifies whether $Auth_s \overset{?}{=} Auth^*_s$ or not. If verification fails, the session is terminated. Otherwise, U_m computes $Auth_{is} = h(ID_m||KS||n_1||n_2))$and sends $\{Auth_{is}\}$ to the server.

Step 5 After receiving the message, server checks if $Auth_{is} \overset{?}{=} h(ID_m||KS||n_1||n_2))$ is true or not. If it holds, then session key is verified, otherwise server terminates the session.

5.5 Password Change Phase

In this phase, a legal user U_m can change his password using the following steps.

Step 1 U_m enters his SC into a card reader, inputs his ID_m, password PW_m, imprints his biometric B_m. Then, computes $PW^*_{nw} = h(PW_m||H(B_m))$, $N^*_m = P_m \oplus h(ID_m||PW^*_{nw})$, $Q^*_m = (ID_m||PW^*_{nw}||N^*_m)$. Now, SC verifies $Q_m \overset{?}{=} Q^*_m$. If the condition is not satisfied, the request is rejected for password change and terminates the session. Otherwise, SC allows the user to enter a new password PW^{new}_m.

Step 2 SC again calculates $PW^{new*}_{nw} = h(PW^{new}_m||H(B_m))$, $N^{new}_m = P_m \oplus h(ID_m||PW^{new}_{nw})$, $Q^{new}_m = (ID_m||PW^{new}_{nw}||N^{new}_m)$. Finally, the parameters $\{P_m, Q_m\}$ are replaced with $\{P^{new}_m, Q^{new}_m\}$ in smart card.

5.6 Smart Card Revocation Phase

The user can revoke his smart card if the smart card is lost or stolen. The user can re-register with the same identity to obtain a new smart card.

Step 1 For revocation of a smart card, U_m keeps the identity and biometric same but chooses a different password PW_d. Then, computes $h(PW_d||H(B_m))$ and sends it to the server along with pseudonym identity PID_m.

Step 2 Upon receiving the message, S verifies the registration of user by checking the user identity. If user ID_m exist, then it sets N=N+1 and computes $\{M_m, P_m, Q_m\}$. Otherwise, it rejects the session.

Step 3 Now, S embedded the computed parameters into the SC and issues it to the U_m. And, updates $N = N + 1$ in its database.

6 Security Analysis of the Proposed Scheme

This section describes the formal security of the proposed scheme. Both BAN logic and random oracle model have been used to prove mutual authentication and session key security. Later, the informal security analysis of the proposed scheme, such as passive and active attacks, are discussed. Also, the proposed scheme achieves mutual authentication, session key security, and user anonymity.

6.1 Authentication Proof Using BAN Logic

BAN logic is widely used to proves the mutual authentication between the user and server [43]. In this section, we proved the authentication between the user and server using BAN logic. Let symbols γ and φ are principals, κ and υ range overstatements, and λ ranges over the cryptographic key. We have taken some notations of the BAN logic as follows:

- $\gamma \mid\equiv \kappa$: γ believes κ.
- $\#(\kappa)$: κ is fresh.
- $\gamma \Rightarrow \kappa$: γ has jurisdiction over κ.
- $\gamma \lhd \kappa$: γ sees κ after receiving it.
- $\gamma \mid\sim \kappa$: Previously γ sent a message including κ.
- $< \kappa >_{\upsilon}$: κ is combined with υ.
- $(\kappa)_h$: κ is hashed with the key λ.
- $(\kappa)_{\lambda}$: κ is encrypted with λ.
- $\gamma \overset{\lambda}{\leftrightarrow} \varphi$: λ is a secret share key between γ and φ. Only γ and φ know about the λ and not others.
- The message meaning rule $\frac{\gamma|\equiv\gamma\overset{\lambda}{\leftrightarrow}\varphi,\gamma\lhd(\kappa)_{\lambda}}{\gamma|\equiv\varphi|\sim\kappa}$
- The nonce verification rule $\frac{\gamma|\equiv\kappa(\kappa),\gamma|\equiv\varphi|\sim\kappa}{\gamma|\equiv\varphi|\equiv\kappa}$
- The jurisdiction rule $\frac{\gamma|\equiv\varphi\Rightarrow\kappa,\gamma|\equiv\varphi|\equiv\kappa}{\gamma|\equiv\kappa}$
- The freshness rule $\frac{\gamma|\equiv\#\kappa}{\gamma|\equiv\#(\kappa,\upsilon)}$
- The belief rule $\frac{\gamma|\equiv\varphi|\equiv(\kappa,\upsilon)}{\gamma|\equiv\varphi|\equiv(\kappa)}$

According to BAN logic, our scheme meets following four goals.

Goal 1: $U_m \mid\equiv U_m \overset{KS}{\leftrightarrow} S_n$

Goal 2: $U_m \mid\equiv S_n \mid\equiv U_m \overset{KS}{\leftrightarrow} S_n$

Goal 3: $S_n \mid\equiv U_m \overset{KS}{\leftrightarrow} S_n$

Goal 4: $S_n \mid\equiv U_m \mid\equiv U_m \overset{KS}{\leftrightarrow} S_n$

The following assumptions has been taken to transform the enhanced scheme to the idealized as follows:

Message 1: $U_m \to S_n \; : \; (ID_m, M_i, n_1, U_m \overset{Z_1}{\leftrightarrow} S_n)_{h(ID_m \| x)}$

Message 2: $S_n \to U_m \; : \; (SID_j, N_m, Y_1, U_m \overset{KS}{\leftrightarrow} S_n)_{h(ID_m \| x)}$

Message 3: $U_m \to S_n \; : \; (ID_m, n_1, n_2, U_m \overset{KS}{\leftrightarrow} S_n)_{(KS)}$

We make some initial state assumptions to analyze the proposed scheme

A_1: $U_m \mid\equiv \#Z_1$

A_2: $S_n \mid\equiv \#Y_1$

A_3: $U_m \mid\equiv (U_m \overset{h(ID_m\|x)}{\leftrightarrow} S_n)$

A_4: $S_n \mid\equiv (U_m \overset{h(ID_m\|x)}{\leftrightarrow} S_n)$

A_5: $U_m \mid\equiv S_n \mid\Rightarrow (U_m \overset{Y_1}{\leftrightarrow} S_n)$

A_6: $S_n \mid\equiv U_m \mid\Rightarrow (U_m \overset{Z_1}{\leftrightarrow} S_n)$

A_7: $U_m \mid\equiv S_n \mid\Rightarrow (U_m \overset{KS}{\leftrightarrow} S_n)$

A_8: $S_n \mid\equiv U_m \mid\Rightarrow (U_m \overset{KS}{\leftrightarrow} S_n)$

The idealized form of our scheme is studied based on the BAN logic and the assumptions. The proofs are as follows:

According to message 1, we have

Step 1 $S_n \lhd (ID_m, M_i, n_1, U_m \overset{Z_1}{\leftrightarrow} S_n)_{h(ID_m\|x)}$

According to Step 1, A_4, we applying message meaning rule to have

Step 2 $S_n \mid\equiv U_m \mid\sim (ID_m, M_i, n_1, U_m \overset{Z_1}{\leftrightarrow} S_n)$

According to Step 2, A_2, we apply the freshness conjuncatenation rule to obtain

Step 3 $S_n \mid\equiv U_m \mid\equiv (ID_m, M_i, n_1, U_m \overset{Z_1}{\leftrightarrow} S_n)$

From Step 3, we apply break conjunctions to produce

Step 4 $S_n \mid\equiv U_m \mid\equiv (U_m \overset{Z_1}{\leftrightarrow} S_n)$

From Step 4, A_6, by applying the jurisdiction rule to get

Step 5 $S_n \mid\equiv (U_m \overset{Z_1}{\leftrightarrow} S_n)$

Session key is computed as $KS = n_2.Z_1 = n_2.n_1.P$. So, we could obtain following Step

Step 6 $S_n \mid\equiv U_m \overset{KS}{\leftrightarrow} S_n$ **(Goal-3)**

According to message 2, we have

Step 7 $U_m \lhd (SID_j, N_m, Y_1, U_m \overset{KS}{\leftrightarrow} S_n)_{h(ID_m\|s)}$

According to Step 7, A_3, and message meaning rule, we get

Step 8 $U_m \mid\equiv S_n \mid\sim (SID_j, Y_1, U_m \overset{KS}{\leftrightarrow} S_n)$

From assumption A_1 and freshness conjuncatenation rule, we obtain

Step 9 $U_m \mid\equiv S_n \mid\equiv (SID_j, Y_1, U_m \overset{KS}{\leftrightarrow} S_n)$

According to Step 9, we apply the BAN logic rule to break the conjunctions

Step 10 $U_m \mid\equiv S_n \mid\equiv U_m \overset{KS}{\leftrightarrow} S_n$ **(Goal-2)**

From Step 10, A_7, and jurisdiction rule, we have

Step 11 $U_m \mid\equiv U_m \overset{KS}{\leftrightarrow} S_n$ **(Goal-1)**

From message 3, we have

Step 12 $S_n \lhd (ID_m, n_1, n_2, U_m \overset{KS}{\leftrightarrow} S_n)_{KS}$

From Step 12, A_8, and message meaning rule, we obtain

Step 13 $S_n \mid\equiv U_m \mid\sim (ID_m, n_1, n_2, U_m \overset{KS}{\leftrightarrow} S_n)$

From assumption A_2 and freshness conjuncatenation rule, we have

Step 14 $S_n \mid\equiv U_m \mid\equiv (ID_m, n_1, n_2, U_m \overset{KS}{\leftrightarrow} S_n)$

According to Step 14, we apply the BAN logic rule to break the conjunctions

Step 15 $S_n \mid\equiv U_m \mid\equiv U_m \overset{KS}{\leftrightarrow} S_n$ **(Goal-4)**

Based on the above analysis, we generalize that both U_m and S_n believe that a session key is shared between them.

6.2 Formal Security Analysis

In this section, we construct the formal security analysis of the proposed scheme based on the random oracle method [44, 45]. The analysis describes the proposed scheme is secure even if the user identity and secret key are revealed. To apply the method of contradiction, we assume that there exist the following two random oracles available for an adversary.

- *Reveal* This random oracle returns the input γ from the output hash value $\varphi = h(\gamma)$.
- *Extract* This random oracle returns the scalar n out of a given point $P = nR$ and R.

Theorem 1 *The proposed scheme is provably secure against an attacker for deriving the user id* ID_m *and the secret key* $\{x, y\}$ *under the hardness assumption of ECDLP and the one-way hash function which behaves like random oracle.*

Proof Consider an adversary A has the ability to derive the ID_m and server's private key x by eavesdropping the login message. An adversary can run the experiment $EXP_{A,UAPS}^{ECDLP,HASH}$ against the proposed user anonymity preserving authentication scheme $UAPS$ by simulating both the oracles *Reveal* and *Extract*.

The success probability of $EXP_{A,UAPS}^{ECDLP,HASH}$ is defined by $|2pr[EXP_{A,UAPS}^{ECDLP,HASH} = 1] - 1|$. The advantage function is defined by $Advt1(t_1, q_e, q_r) = max\{succ1\}$, where A can take maximum execution time t_1 and can make maximum q_e extract, q_r reveal queries. The proposed scheme can capable to calculate ID_m and secret key $\{x, y\}$, if $Advt1(t_1, q_e, q_r) \leqslant \varepsilon$ for any small $\varepsilon \geqslant 0$. By using Definitions 1 and 2, to break a oneway hash function and ECDLP is an infeasible work for an adversary. Hence, the theorem is proved. □

Algorithm 1 $EXP_{A,UAPS}^{ECDLP,HASH}$

```
Assume the login message $\{UID_m, Z_3, Z_4\}$ has been intercepted during the login phase
  of proposed scheme, where
  $Z_3 = n_1 \oplus M_i^*$, $Z_4 = h(ID_m \| M_i^* \| N_m^* \| n_1 \| Z_1))$
Call Reveal 1 on input $Z_4$ and get $(ID'_m \| M_i^{*'} \| N_m^{*'} \| n'_1 \| Z'_1) \leftarrow$ Reveal $(Z_4)$
Call Extract 1 on $M_i^{*'}$ and get $(ID_m^* \| H(B_m)) \leftarrow$ Reveal $(M_i^*)$
if $(ID'_m = ID_m^*)$ then
    Compute $n_1^* = Z_3 \oplus M_i^{*'}$
    if $(n'_1 = n_1^*)$ then
        Compute $Z'_4 = h(ID'_m \| M_i^* \| N_m^* \| n_1^* \| Z'_1)$
        Call Reveal 1 on $N_m^{*'}$ and get $(ID_m^{**} \| x) \leftarrow$ Reveal $(N_m^{*'})$
        if $(Z_4 = Z'_4)$ then
            Accept $ID_m^{**}$ and $x$
            return 1
            else
            return 0
        end if
        else
        return 0
    end if
    else
    return 0
end if
```

6.3 Security Analysis Against Other Possible Attacks

This section presents the informal security analysis of the proposed scheme.

6.3.1 Key Compromise Impersonation Attack

Let an adversary eavesdrops the login message and also the secret key x and y are compromised. Even if $\{UID_m, PID_m, Z_3, Z_4\}$ are sent in public channel, and the secret key is known, an adversary will not be able to verify Z_4. For the validation of Z_4, adversary needs ID_m, M_i, n_1 and N_m, where $M_i = h(ID_m \| H(B_m)).P = (P_x, P_y)$, $Z_1 = n_1.P$ and n_1 is a random number generated by the user. Hence, the proposed scheme can resist key compromise impersonation attack.

6.3.2 Known Session-Specific Temporary Information Attack

The proposed scheme successfully resist this attack. Even if an adversary knows the temporary random number $n1$, he could not compute the session key without the knowledge of S_1, T_1, N_m, where $S_1 = n_2.Z_1^*$, $T_1 = n_1.Y$, and $N_m = h(ID_m \| x)$. Moreover, Y_1 is sent by encrypted form, and N_m is computed by using user identity and the server's private key. Thus, the proposed scheme can resist known session-specific temporary information attack.

6.3.3 Lost Smart-Card Attack

Suppose an adversary can get the user's smart card. He can easily extract the parameters $\{P_m, Q_m, UID_m, PID_m, h(.), E_k, D_k\}$ using the power analysis. Still, he can not derive any further information from $P_m = N_m \oplus h(ID_m \| PW_{nw})$, $Q_m = (ID_m \| PW_{nw} \| N_m)$ because they are protected by one way hash function and secret key x. In addition, the attacker cannot guess ID_m, PW_m, x at the same time. Thus, our scheme could withstand a lost smart card attack.

6.3.4 Known Key Security

The session key of the proposed protocol $KS = h(S_1 \| N_m)$ is depends on the nonce n_1 and n_2 generated by U_m and S respectively. As the nonce is generated in each session freshly, so the session key will be different for each session. Hence, the compromise of one session key will not be an advantage of computing another session key. Thus, the proposed scheme achieves known key security.

6.3.5 User Anonymity

User anonymity intends to preserve the secrecy of the user identity throughout the communication. In the enhanced scheme, the login message, and the smart card information does not contain user ID_m in plain text. The messages sent through the public and private channels are protected by the collision-resistant one-way hash function, from which user identity could not be retrieved. Thus, our scheme achieves user anonymity.

6.3.6 Perfect Forward Secrecy

In the proposed protocol, the session key KS is computed as $KS = h(S_1 \| N_m)$, where $S_1 = n_2.Z_1^*$, $N_m = h(ID_m \| x)$. Even if the secret key x is revealed, an adversary could not compute the session key because of intractability of Diffie-Hellman problem. Hence, our scheme could provide perfect forward secrecy.

6.3.7 Stolen Verifier Attack

In the stolen verifier attack, an adversary can read user ID_m, password PW_m, and biometric B_m stored in the verification table at the server. After getting the ID_m, PW_m, and B_m, the adversary acts as a valid user. In the proposed scheme, the PW_m and $H(B_m)$ have not been stored in the verification table. From ID_m, an adversary cannot obtain any information. Hence, the proposed scheme can resist a stolen verifier attack.

6.3.8 User Unlinkability

User unlinkability means no adversary can distinguish whether the two different sessions are initiated by the same user. However, in the proposed protocol, the login message computed as $Z_3 = n_1 \oplus M_i^*$, $Z_4 = h(ID_m \| M_i^* \| N_m^* \| n_1 \| Z_1)$, where n_1 is the random number generated by the user. Thus, the login message will be different in each session. Although the adversary gets the login message, he could not verify whether two login messages are from the same user or not. So, the proposed scheme preserves user unlinkability.

6.3.9 Efficient Login Phase

In the login phase, the smart card verifies the legitimacy of a user by using its stored information. When the user inserts his identity, password, and imprints his biometric, smart card computes $PW_{nw} = h(PW_m \| H(B_m))$, $N_m^* = P_m \oplus h(ID_m \| PW_{nw}^*)$, $Q_m^* = (ID_m \| PW_{nw}^* \| N_m^*)$. Then, it verifies the condition $Q_m \overset{?}{=} Q_m^*$. If the condition does not satisfy, the smart card terminates the session. Otherwise, the user is a valid user. The *SC* validates the user first and then sends the login message to the server. Thus, the proposed scheme has an efficient login phase.

6.3.10 Mutual Authentication

In our scheme, the user and server authenticated each other as follows.

After obtaining the login message $\{UID_m, PID_m, Z_3, Z_4\}$ from U_m, the server computes n_1^*, Z_1^*, and Z_4^*. Then, the server compares the computed Z_4^* with the received Z_4 to check for the authenticity of the user. If the condition fails, the server aborts the session. Otherwise, computes the parameters $\{Y_2, Auth_s\}$, and sends it to the user. After receiving the authentication message, the user will first compute $Auth_s^*$ and matches with the received $Auth_s$. If both are equal, then the user will verify the server, otherwise rejects the session.

7 Simulation of Proposed Scheme Using AVISPA

This section demonstrates the simulation of the proposed scheme using Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [46]. It is a push-button tool which analyses and validates security protocols automatically. AVISPA is a modular and expressive formal language for specifying protocols and their security properties. The protocol defined in the High-Level Protocol Specification (HLPSL) and translated into Intermediate Format (IF) using HLPSL2IF translator [47]. There are four back-ends that are OFMC, CL-Atse, SATMC, and TA4sp. The output of IF is used as input to the back-ends and produce the output format (OF). The assumption is the transmission channel of the HLPSL is controlled by the Dolev–Yao model. The structure of the AVISPA tool is presented in Fig. 1.

7.1 Specifying the Scheme

This section demonstrates the four phases of our scheme using the HLPSL language. There are two basic roles user (U_m) and server (S). The role of the (U_m) first receives the start signal and changes its state from 0 to 1. Then, U_m sends the registration message $\{ID_m, PW_{nw}, H(B_m)\}$ to the S through the secure channel using SND() operation and receives the smart card having the information $\{P_i, Q_i, E_k, D_k, h(.)\}$ from the S using RCV()

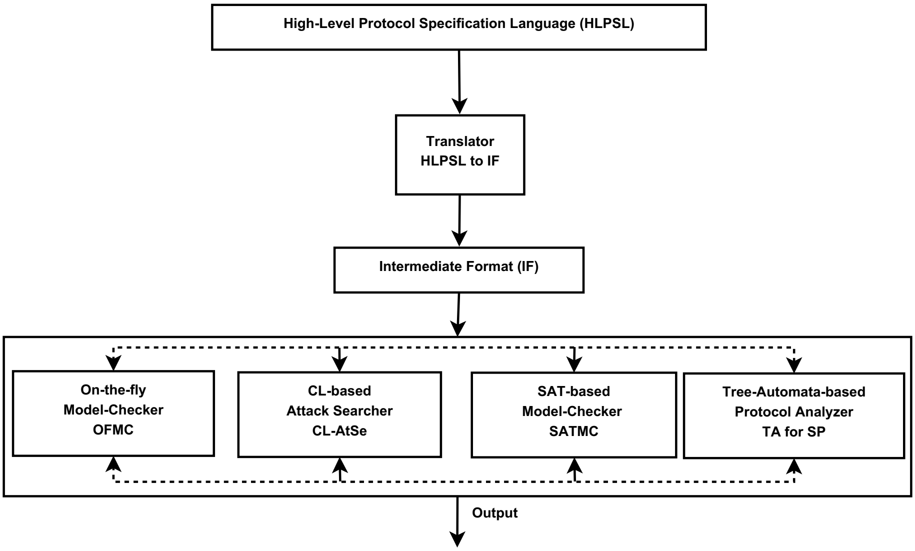


Fig. 1 The architecture of the AVISPA tool

operation. During the login phase, the user sends the login request message $\{UID_m, Z_3, Z_4\}$ to the server through a public channel. Finally, the user receives the authentication message $\{Y_2, Au\}$ from the server, and sends $\{Aui\}$ to complete mutual authentication. As channel (dy) declares insecure, an intruder can insert, modify, or delete the message during the communication.

The declaration witness $\{S, A, auth\_a\_s\_Aui, Aui'\}$ expresses a weak authentication property, which means the user has freshly generated the value of $Z1'$ for the server. The declaration request $\{A, S, auth\_s\_a\_Au, Au1'\}$ shows a strong authenticated which intends user's acceptance of value $Au1'$ generated for the user by a server.

The declaration type secret secret $(\{Pwi\}, sec1, peke_A)$ depicts that the information PW_m is kept secret to the user U_m only and characterized by the protocol id $sec1$. The goal secrecy expresses the variable V is kept permanently secret. The role specification of U_m and S are given in Tables 6 and Table 7 respectively. Table 8 presents the role specification of session, goal, and environment of the proposed scheme.

The simulation result of OFMC and CL-Atse background is shown in Fig. 2a and b respectively.

8 Performance Evaluation

This section demonstrates the comparison of related existing schemes and the proposed scheme in terms of computational cost, communicational cost, and security features. Table 9 shows the computational cost analysis in which T_{HS}, T_{EL}, T_{IN}, and T_{EM} denote

Table 6 Role of user

```
role peke_A(A, S : agent,
Snd, Rcv :channel (dy),
Snd1,Rcv1 :channel(ota),
Xu : symmetric_key,
Hash,XOR, Mul, Enp, Dep: function)
played_by A def=
local State :nat,
IDi, PWi, Bm, UIDi, N1, N2 :text,
Pwi1, Ni, Mi, Y2, P:message,
KS: symmetric_key,
Pub: public_key,
N11, Au1, Au, Aui, G1, Y1: message,
Sdd: message,
Pwill,Pi,Qi,Qi1,Z1, Z2, Z3, Z4,Dp :message
sec1,sec2:protocol_id
init State:=0
transition
0.State =0 /\Rcv1(start)= >
State’ :=1
/\Pwi1’:=Hash(Pwi.Hash(Bm’))
/\Snd1(IDi, Pwil’, Hash(Bm’))
/\secret({Pwi},sec1,peke_A)
1.State =1 /\Rcv1(Pi,Qi1,UIDi)= >
State’:=2
/\ Pwi1’:=Hash(Pwi’.Hash(Bm’))
/\ Ni’:=XOR(Pi’,Hash(IDi’.Pwi1’))
/\ Qi’:= Hash(IDi’.Pwi1’.Ni’)
/\ N1’:=new()
/\ Mi’:=Mul(Hash(IDi’.Hash(Bm’)).P’)
/\ Z1’:=Mul(N1’.P’)
/\Z2’:=Mul(N1’.Pub)
/\Z3’:=XOR(N1’,Mi’)
/\Z4’:=Hash(IDi’.Mi’.Ni’.N1’.Z1’)
/\Snd(UIDi,Z3’,Z4’)
1.State =2 /\ Rcv(Y2,Au)= >State’:=3
/\ Y1′:=Dep(Y2)
/\ N2′:=Dep(Y2)
/\G1’:= Mul(N1’.Y1’)
/\KS’:=Hash(G1.Ni’)
/\Au1’:=Hash(Sdd’.KS’.Ni’.Y1’)
/\Aui’:=Hash(IDi’.KS’.N1’.N2’)
2.State = 3 /\Snd(Aui’) = >State’ :=3
/\request(A, S, auth_a_s_qi, Qi’)
/\ witness(S,A, auth_s_a_Z4, Z4’)
/\request(A, S, auth_a_s_Au, Au1’)
/\witness(S,A, auth_s_a_Aui, Aui’)
end role
```

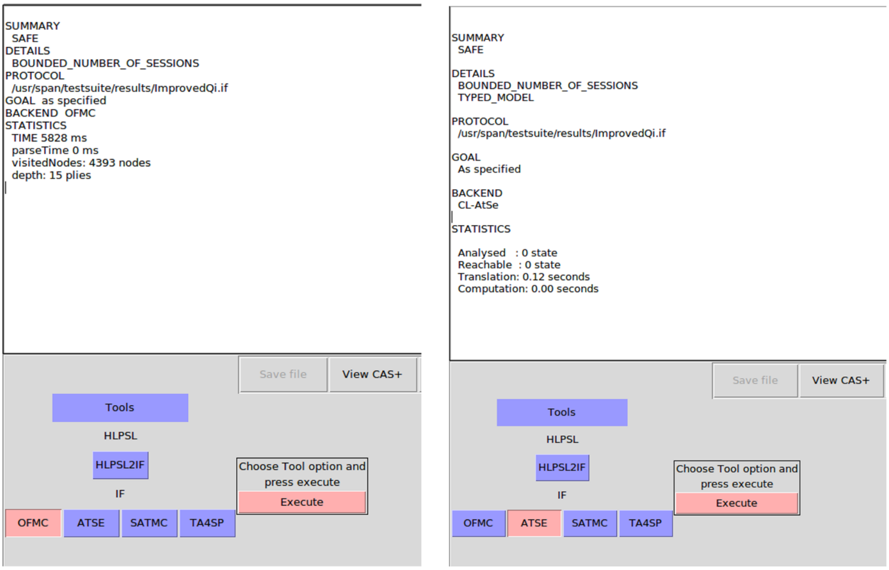
Table 7 Role of server

```
role peke_S(A, S : agent,
Snd, Rcv :channel (dy),
Snd1,Rcv1: channel(ota),
Xu: symmetric_key,
Hash,XOR, Mul, Enp, Dep : function)
played_by S def=
local State :nat,
IDi, PWi, Bm, UIDi, N1, N2 :text,
Pwi1, Ni, Mi, Y2, P:message,
KS: symmetric_key,
Pub: public_key,
N11, Au1, Au, Aui, G1, Y1: message,
Sdd: message,
Pwill, Pi, Qi, Qi1, Z1, Z2, Z3, Z4,Dp :message
sec1,sec2:protocol_id
init State:=0
transition
0.State =0 /\Rcv1(IDi,Pwi1,Hash(Bm))= >
State’ :=1
/\Xu’:=new()
/\Mi1’:= Mul(Hash(IDi’.Hash(Bm’)).P’)
/\Ni1’:=Hash(IDi’.Xu’)
/\Pi’:=XOR(Ni1’,Hash(IDi’.Pwill’))
/\Qi1’:=Hash(IDi’.Pwil’.Ni1’)
1.State =1 /\Snd1(Pi’,Qi1’,UIDi)= >
/\secret({Xu},sec2,peke_S) State’ :=2
/\ Rcv(Z3,Z4)
/\ Ni1’:=XOR(Z3’,Mi1’)
/\ Z1’:=Mul(N1’.P’)
/\ Z4’:= Hash(IDi.Mi’.Ni’.N1’.Z1’)
/\ N2’:=new()
/\ Y1’:=Mul(N2’.P’)
/\ Y2’:=Enp{N2′.Y1′}
/\G1’:=N2’.Z1’
/\KS’:=Hash(G1’.Ni1’)
/\Au’:= Hash(Sdd’.KS’.Ni1’.Y1’)
/\Snd(Y2’,Au’)
1.State =2 /\ Rcv(Aui) = >
State’ :=2
/\ Auii’:=Hash(IDi.KS’.N1’.N2’)
/\witness(A, S, auth_a_s_qi, Qi1’)
/\ request(S,A, auth_s_a_Z4, Z41’)
/\witness(A, S, auth_a_s_Au, Au’)
/\request(S,A, auth_s_a_Aui, Auii’)
end role
```

Table 8 Role environment

```
role session (A, S : agent,
Xu: symmetric_key,
Hash,XOR, Mul: function)
def=
local A_SND, A_RCV, S_SND, S_RCV:channel (dy),
A_SND1, A_RCV1, S_SND1, S_RCV1:channel(ota)
composition
peke_A(A,S,A_SND, A_RCV, A_SND1, A_RCV1, Xu, Hash, XOR, Mul)
/\peke_S(A,S,S_SND, S_RCV, S_SND1, S_RCV1, Xu, Hash, XOR, Mul)
end role
role environment()
def=
const
a, s, i: agent,
xu, pwi :symmetric_key,
hhash,xorr,mul,enp,dep :function,
auth_a_s_qi :protocol_id,
auth_s_a_Z4 :protocol_id,
auth_a_s_Au: protocol_id,
auth_s_a_Aui:protocol_id
sec1,sec2:protocol_id
intruder_knowledge=i,a,s,pwi,hhash,xorr,mul
composition
session(a,s,xu, hhash, xorr, mul)
/\ session(a,s,xu, hhash, xorr, mul)
/\ session(i,s,pwi, hhash, xorr, mul)
/\ session(a,i,pwi, hhash, xorr, mul)
/\ session(i,s,pwi, hhash, xorr, mul)
end role
goal
secrecy_of sec1
secrecy_of sec2
authentication_on auth_a_s_qi
authentication_on auth_s_a_Z4
authentication_on auth_a_s_Au
authentication_on auth_s_a_Aui
end goal
environment()
```

hash function, elliptic curve point, inverse function, and encryption/decryption function respectively. The total computational cost of our scheme is $15T_{HS} + 7T_{EL} + 2T_{EM}$, which is somewhat more than other existing schemes. We have implemented all operation tate_ bilinear_pairing eta and tate_bilinear_pairing ecc package in Python library. The experiment carried on using a laptop running Windows 10 and 64-bit Intel(R) Core(TM) i3 CPU M380 @2.53 GHz, 4.00 GB RAM. Since the running time of the exclusive-OR operation



```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /usr/span/testsuite/results/ImprovedQi.if
GOAL  as specified
BACKEND  OFMC
STATISTICS
  TIME 5828 ms
  parseTime 0 ms
  visitedNodes: 4393 nodes
  depth: 15 plies
```

(a) Result using OFMC backend

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /usr/span/testsuite/results/ImprovedQi.if

GOAL
  As specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 0 state
  Reachable  : 0 state
  Translation: 0.12 seconds
  Computation: 0.00 seconds
```

(b) Result using ATSE backend

Fig. 2 Simulation result result using AVISPA tool

Table 9 Analysis of computational cost

Scheme	User	Server	Total cost
Lu et al. [35]	$6T_{HS} + 2T_{EL}$	$6T_{HS} + 2T_{EL}$	$12T_{HS} + 4T_{EL} \approx 0.248s$
Chaudhry et al. [36]	$6T_{HS} + 4T_{EL}$	$4T_{HS} + 2T_{EL} + 1T_{IN}$	$10T_{HS} + 6T_{EL} + 1T_{IN} \approx 0.3689s$
Wu et al. [29]	$8T_{HS} + 2T_{EL} + 2T_{EM}$	$8T_{HS} + 2T_{EL} + 2T_{EM}$	$16T_{HS} + 4T_{EL} + 4T_{EM} \approx 5.308s$
Qi et al. [39]	$11T_{HS} + 3T_{EL} + 1T_{EM}$	$5T_{HS} + 3T_{EL} + 1T_{EM}$	$16T_{HS} + 6T_{EL} + 2T_{EM} \approx 2.898s$
Proposed scheme	$10T_{HS} + 4T_{EL} + 1T_{EM}$	$5T_{HS} + 3T_{EL} + 1T_{EM}$	$15T_{HS} + 7T_{EL} + 2T_{EM} \approx 2.956s$

Table 10 Analysis of communicational cost

Scheme	Message transfer	Commu-nicational cost
Lu et al. [35]	3	1376 bits
Chaudhry et al. [36]	2	1344 bits
Wu et al. [29]	2	1152 bits
Qi et al. [39]	3	1344 bits
Proposed scheme	3	960 bits

is negligible, the computation cost of EX-OR function is omitted. Compared to the security features, the increase in computational is acceptable.

Table 10 compares the message exchange and communicational cost of our scheme with other related schemes. The message exchange in Lu et al. and Qi et al. is three whereas

Table 11 Comparison of security features

Scheme	SF_1	SF_2	SF_3	SF_4	SF_5	SF_6	SF_7	SF_8	SF_9
Lu et al. [35]	×	×	✓	✓	✓	×	✓	✓	×
Chaudhry et al. [36]	×	×	✓	✓	✓	×	✓	✓	×
Wu et al. [29]	✓	✓	✓	✓	×	×	✓	✓	×
Qi et al. [39]	×	×	✓	✓	×	✓	✓	✓	×
Proposed scheme	✓	✓	✓	✓	✓	✓	✓	✓	✓

SF_1—user anonymity, SF_2—key compromise impersonation attack, SF_3—lost smart card attack, SF_4—known key security, SF_5—perfect forward secrecy, SF_6—user unlinkability, SF_7—efficient login phase, SF_8—mutual authentication, SF_9—smart card revocation

Chaudhry et al. and Wu et al. is two. The proposed scheme also needs three message exchange between user and server in the login and authentication phase. For the communicational cost, the assumption is the length of the identity, length of the nonce/time stamp is 32 bits, length of the encryption/hash function is 160 bits, and elliptic curve point is 320 bits. With these values, the communicational cost of Lu et al., Chaudhry et al.,Wu et al., and Qi et al. are 1376, 1344, 1152, and 1344 bits respectively. The communication cost of the proposed scheme is 960 bits which is less than other existing schemes.

Table 11 manifests the functionality features of the proposed scheme with other related schemes. Both of Wu et al. and Qi et al. schemes could not achieve perfect forward secrecy. In addition, Lu et al., Chaudhry et al., and Qi et al. schemes are vulnerable to key compromise impersonation attack and could not achieve user anonymity. Also, Lu et al., Chaudhry et al., and Wu et al. schemes are fail to provide user unlinkability. The proposed scheme is considerably more secure and fulfills the desirable security features. Also, the proposed scheme achieves the extra feature that is smart card revocation for which the user can re-register if the smart card lost or stolen.

9 Conclusion

In this paper, we have reviewed Qi et al. 's scheme and show that their scheme is susceptible to key compromise impersonation attack, offline password guessing attack, and known session-specific temporary information attack. To overcome these flaws, we have proposed a biometric-based authentication scheme for the client-server environment using ECC. We proved the mutual authentication of our scheme using BAN logic and session key security through ROM. Further, the formal verification of the proposed scheme using the AVISPA tool shows the scheme is secure. In addition, the informal security analysis demonstrates that the scheme is secure against several known attacks. Though the computational cost of the scheme is a little bit more, the security and performance analysis depicts that our scheme is secure and suitable for practical application.

References

1. Hwang, T., Chen, Y., & Laih, C. J. (1990). Non-interactive password authentications without password tables. In *1990 IEEE region 10 conference on computer and communication systems, 1990. IEEE TENCON'90* (pp. 429–431). IEEE.

2. Yang, W.-H., & Shieh, S.-P. (1999). Password authentication schemes with smart cards. *Computers & Security*, *18*(8), 727–733.
3. Pippal, R. S., Jaidhar, C. D., & Tapaswi, S. (2013). Robust smart card authentication scheme for multi-server architecture. *Wireless Personal Communications*, *72*(1), 729–745.
4. Wei, J., Liu, W., & Xuexian, H. (2014). Cryptanalysis and improvement of a robust smart card authentication scheme for multi-server architecture. *Wireless Personal Communications*, *77*(3), 2255–2269.
5. Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM*, *24*(11), 770–772.
6. Chang, C.-C., & Wu, T.-C. (1991). Remote password authentication with smart cards. *IEE Proceedings E-Computers and Digital Techniques*, *138*(3), 165–168.
7. Hwang, M.-S., & Li, L.-H. (2000). A new remote user authentication scheme using smart cards. *IEEE Transactions on Consumer Electronics*, *46*(1), 28–30.
8. Fan, L., Li, J.-H., & Zhu, H.-W. (2002). An enhancement of timestamp-based password authentication scheme. *Computers & Security*, *21*(7), 665–667.
9. Lin, C.-W., Tsai, C.-S., & Hwang, M.-S. (2006). A new strong-password authentication scheme using one-way hash functions. *Journal of Computer and Systems Sciences International*, *45*(4), 623–626.
10. Chan, C.-K., & Cheng, L.-M. (2000). Cryptanalysis of a remote user authentication scheme using smart cards. *IEEE Transactions on Consumer Electronics*, *46*(4), 992–993.
11. Sun, H.-M. (2000). An efficient remote use authentication scheme using smart cards. *IEEE Transactions on Consumer Electronics*, *46*(4), 958–961.
12. Yoon, E.-J., Ryu, E.-K., & Yoo, K.-Y. (2004). Further improvement of an efficient password based remote user authentication scheme using smart cards. *IEEE Transactions on Consumer Electronics*, *50*(2), 612–614.
13. Rongxing, L., & Cao, Z. (2005). Efficient remote user authentication scheme using smart card. *Computer Networks*, *49*(4), 535–540.
14. Lee, S.-W., Kim, H.-S., & Yoo, K.-Y. (2005). Improvement of Chien et al.'s remote user authentication scheme using smart cards. *Computer Standards & Interfaces*, *27*(2), 181–183.
15. Lee, N.-Y., & Chiu, Y.-C. (2005). Improved remote authentication scheme with smart card. *Computer Standards & Interfaces*, *27*(2), 177–180.
16. Jing, X., Zhu, W.-T., & Feng, D.-G. (2009). An improved smart card based password authentication scheme with provable security. *Computer Standards & Interfaces*, *31*(4), 723–728.
17. Sahoo, S. S., Mohanty, S., & Majhi, B. (2018). An improved and secure two-factor dynamic id based authenticated key agreement scheme for multiserver environment. *Wireless Personal Communications*, *101*, 1307–1333.
18. Guo, D., & Wen, F. (2014). Analysis and improvement of a robust smart card based-authentication scheme for multi-server architecture. *Wireless Personal Communications*, *78*(1), 475–490.
19. Li, X., Niu, J., Kumari, S., Liao, J., & Liang, W. (2015). An enhancement of a smart card authentication scheme for multi-server architecture. *Wireless Personal Communications*, *80*(1), 175–192.
20. Tan, Z. (2016). A privacy-preserving multi-server authenticated key-agreement scheme based on Chebyshev chaotic maps. *Security and Communication Networks*, *9*(11), 1384–1397.
21. Li, C.-T., & Hwang, M.-S. (2010). An efficient biometrics-based remote user authentication scheme using smart cards. *Journal of Network and Computer Applications*, *33*(1), 1–5.
22. Sahoo, S. S., Mohanty, S., & Majhi, B. (2017). A light weight three factor based authentication scheme for multi-server environment using smart cards. In *Proceedings of the 2017 the 7th international conference on communication and network security* (pp. 43–47). ACM.
23. Sureshkumar, V., Amin, R., & Anitha, R. (2017). An enhanced bilinear pairing based authenticated key agreement protocol for multiserver environment. *International Journal of Communication Systems*, *30*(17), e3358.
24. Irshad, A., Sher, M., Chaudhry, S. A., Saru Kumari, Q. X., & Wu, F. (2018). An improved and secure chaotic map based authenticated key agreement in multi-server architecture. *Multimedia Tools and Applications*, *77*(1), 1167–1204.
25. Chaudhry, S. A., Naqvi, H., Farash, M. S., Shon, T., & Sher, M. (2018). An improved and robust biometrics-based three factor authentication scheme for multiserver environments. *The Journal of Supercomputing*, *74*(8), 3504–3520.
26. Sahoo, S. S., & Mohanty, S. (2018). A lightweight biometric-based authentication scheme for telecare medicine information systems using ECC. In *9th international conference on computing, communication and networking technologies (ICCCNT)* (pp. 1–6), IEEE.

27. Yoon, E.-J., & Yoo, K.-Y. (2013). Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. *The Journal of Supercomputing*, *63*(1), 235–255.
28. Yeh, H.-L., Chen, T.-H., Kuei-Jung, H., & Shih, W.-K. (2013). Robust elliptic curve cryptography-based three factor user authentication providing privacy of biometric data. *IET Information Security*, *7*(3), 247–252.
29. Fan, W., Lili, X., Kumari, S., & Li, X. (2015). A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client-server networks. *Computers & Electrical Engineering*, *45*, 274–285.
30. Kim, H., Jeon, W., Lee, K., Lee, Y., & Won, D. (2012). Cryptanalysis and improvement of a biometrics-based multi-server authentication with key agreement scheme. In *International conference on computational science and its applications* (pp. 391–406). Springer.
31. He, D., & Wang, D. (2015). Robust biometrics-based authentication scheme for multiserver environment. *IEEE Systems Journal*, *9*(3), 816–823.
32. Odelu, V., Das, A. K., & Goswami, A. (2015). A secure biometrics-based multi-server authentication protocol using smart cards. *IEEE Transactions on Information Forensics and Security*, *10*(9), 1953–1966.
33. Tan, Z. (2014). A user anonymity preserving three-factor authentication scheme for telecare medicine information systems. *Journal of Medical Systems*, *38*(3), 16.
34. Arshad, H., & Nikooghadam, M. (2014). Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems. *Journal of Medical Systems*, *38*(12), 136.
35. Yanrong, L., Li, L., Peng, H., & Yang, Y. (2015). An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem. *Journal of Medical Systems*, *39*(3), 32.
36. Chaudhry, S. A., Mahmood, K., Naqvi, H., & Khan, M. K. (2015). An improved and secure biometric authentication scheme for telecare medicine information systems based on elliptic curve cryptography. *Journal of Medical Systems*, *39*(11), 175.
37. Mir, O., & Nikooghadam, M. (2015). A secure biometrics based authentication with key agreement scheme in telemedicine networks for e-health services. *Wireless Personal Communications*, *83*(4), 2439–2461.
38. Chaudhry, S. A., Naqvi, H., & Khan, M. K. (2018). An enhanced lightweight anonymous biometric based authentication scheme for TMIS. *Multimedia Tools and Applications*, *77*(5), 5503–5524.
39. Qi, M., & Chen, J. (2018). New robust biometrics-based mutual authentication scheme with key agreement using elliptic curve cryptography. *Multimedia Tools and Applications*, *77*, 1–17.
40. Kocher, P., Jaffe, J., & Jun, B. (1999). Differential power analysis. In M. Wiener (Ed.), *Advances in cryptology 'CRYPTO'99* (pp. 789–789). Berlin: Springer.
41. Messerges, T. S., Dabbish, E. A., & Sloan, R. H. (2002). Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, *51*(5), 541–552.
42. Dolev, D., & Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on Information Theory*, *29*(2), 198–208.
43. Burrows, M., Abadi, M., & Needham, R. M. (1989). A logic of authentication. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, *426*, 233–271.
44. Bellare, M., & Rogaway, P. (1993). Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security* pp. 62–73. ACM.
45. Bresson, E., Chevassut, O., & Pointcheval, D. (2003). Security proofs for an efficient password-based key exchange. In *Proceedings of the 10th ACM conference on computer and communications security* pp. 241–250. ACM.
46. Armando, A., Basin, D., Cuellar, J., Rusinowitch, M., & Viganò, L. (2006, January). Avispa: automated validation of internet security protocols and applications. *ERCIM News*, vol. 64.
47. Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., llar, J. C, Drielsma, P. H., et al. (2005). The AVISPA tool for the automated validation of internet security protocols and applications. In *International conference on computer aided verification* pp. 281–285. Springer.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Shreeya Swagatika Sahoo received M. Tech degree in Computer Science and Engineering from Veer Surendra Sai University, Burla in 2015. Currently, she is pursuing Ph.D. in the Department of CSE at NIT, Rourkela, India. Her current research interests include cryptography, security protocol, and network security.

Sujata Mohanty received Ph.D. degree in Computer Science and Engineering from NIT Rourkela, India. She is currently working as an Assistant Professor in the Department of Computer Science and Engineering at National Institute of Technology, Rourkela, India. Her research interests include information security, cryptography, and network security.

Banshidhar Majhi received his Ph.D. degree from Sambalpur University, Odisha, India, in 2001. He is currently working as a Professor in the Department of Computer Science and Engineering at National Institute of Technology, Rourkela, India. His field of interests include image processing, data compression, cryptography and security, parallel computing, soft computing, and biometrics. He is a professional member of MIEEE, FIETE, LMCSI, IUPRAI, and FIE. He has authored more than hundred papers in journals and conferences of international repute.