

# A Secure Three-Factor Remote User Authentication Scheme Using Elliptic Curve Cryptosystem

Rifaqat Ali and Arup Kumar Pal

**Abstract** Recently, three factors such as biometric, smart card, and password based authentication schemes have drawn considerable attention in the field of information security. In this paper, the authors have presented an elliptic curve cryptosystem based authentication scheme using biometric, smart card, and password and also analyzed the formal and informal security of the authentication scheme. In this scheme, the parameters of elliptic curve are derived from the biometric features like iris, fingerprints, etc., which is suitable to withstand the forgery. The formal and informal security analysis are done based on the BAN logic and suggested propositions, respectively. The security analysis ensures that the presented scheme can withstand various kinds of malicious attacks. In addition, the scheme is also comparable with other related schemes in the context of communication cost, computation cost, and smart card storage. The scheme is suitable to ensure high degree of security with reduced comparatively overhead.

**Keywords** Authentication · BAN logic · Biometric · Key agreement  
Elliptic curve cryptography (ECC) · Smart card

## 1 Introduction

In recent, the e-commerce and m-commerce based applications are become widely popular among users due to the rapid advancement of Internet technology, computer devices, smart phones, etc. Password based authentication is one of the essential security mechanisms during secure communication with these

---

R. Ali (✉) · A. K. Pal  
Department of Computer Science and Engineering,  
Indian Institute of Technology (Indian School of Mines),  
Dhanbad 826004, Jharkhand, India  
e-mail: rifaqatali27@gmail.com

A. K. Pal  
e-mail: arupkrpal@gmail.com

e-commerce and m-commerce applications. In 1981, Lamport [1] presented the first remote user authentication scheme for insecure network. In his scheme, server maintains a password table to authenticate the legitimate user. Since then, in order to improve the system security, computation, and communication efficiency, a large number of smart card and password based authentication have been presented in the literature [2–5]. However, the security flaws in password authentication based protocols have been exposed seriously due to the management of password in improper way. One of the common issues in password based applications is to select suitable password. The selection of long and random password is highly secured but such type of password is not practically convenient to remember for a use. Sometimes, it may happen that the user may share his password with the other people, in that scenario; there is no way to identify who is the legal user. In order to resolve the single password authentication problems, several biometric-based remote user authentication have been presented by several researcher [6–10]. Generally, biometric based remote user authentication is extremely more secure and reliable than the traditional authentication scheme. The advantageous of using biometric keys over the traditional password are like biometric keys cannot be lost or forgotten and even it is not possible to copy, share, and guess the biometric key. The biometric system is basically a pattern recognition system which operates by obtaining biometric data from an individual extracting a feature set from the obtained data and comparing these features with the template set in the database.

In order to design a secure and efficient authentication protocols, many researchers have considered several cryptographic techniques such as *ECC*, *RSA*, non-invertible hash function, and several other mathematical operations such as *XOR* and concatenate. *ECC* provides same level of security with smaller key size than *RSA* (1024-bits *RSA* key is equal to the 160-bits *ECC* key). In 2012, Li [11] presented a two-factor remote user authentication scheme based on *ECC* and claimed that presented protocol is secure against various kinds of security attacks and provides mutual authentication and user anonymity with low computation cost. However, Zhang et al. [12] point out that Li's protocol cannot provide mutual authentication and propose an improved scheme based on *ECC*. They claimed that their scheme provide all security attributes with lower computation cost. In recent years, many *ECC* based mutual authentication and key agreement scheme have been presented in the literature [13–15]. In order to improve the security of remote authentication scheme *ECC* combines with biometric. In 2014, Arshad and Nikooghadam [16] presented *ECC* based three-factor remote authentication and key agreement scheme, which is improvement of Tan et al. scheme [17]. They claimed that their protocol resists various kinds of security flaws with better complexity. Last few years, many biometric and *ECC* based authentication scheme have been presented in the literature for distinct application systems [18, 19]. In this paper, we have also presented of an *ECC* based three-factor mutual authentication scheme where this scheme is verified through formal and informal security analysis.

Rest of the paper is described as follows: In Sect. 2, the *ECC* based three-factor authentication scheme is presented. The security validation using BAN logic

demonstrates in Sect. 3. Moreover, Sect. 4 shows informal security analysis of the presented scheme. The performance comparison is presented in Sect. 5. Finally, the conclusion is drawn in Sect. 6.

## 2 ECC-Based Three-Factor Authentication Scheme

This section presents the three factor authentication scheme based on elliptic curve cryptosystem. The authentication scheme consists of four phases namely registration phase, login phase, authentication and key agreement phase, and password change phase.

### 2.1 Registration Phase

In this phase, a user  $U_i$  sends a request to the authentication server for registration or re-registration purpose. Initially, the user freely chooses his/her identity  $ID_i$ , password  $PW_i$ , and also imprints his/her personal biometrics  $F_i$  at the sensor. Then user calculates  $RPW_i = h(PW_i || r_i)$  and submits  $\{ID_i, RPW_i, F_i\}$  to the server through secure channel. Here  $r_i$  is considered as a random number generated by the user. The server performs the following operations after receiving the message from the user:

- i. Firstly, the server finds out the coordinate points ( $x$ ,  $y$ , and angle) from biometric feature  $F_i$ . From this value, the coefficient of the elliptic curve coordinates value  $A$ ,  $B$ , and  $G$  points [10] are derived.
- ii. The server computes  $T_i = H(F_i)$ , where  $H$  is biohashing function.
- iii. Next, the server computes  $C_i = X \cdot G$ ,  $D_i = h(ID_i || RPW_i) \cdot G$ ,  $E_i = C_i + D_i$ ,  $S_i = C_i + h(RPW_i) \cdot G$ , where  $X$  is master key of the server.
- iv. Finally, the server issues a smart card which contains  $\{G, E_i, H(\cdot), h(\cdot), S_i, T_i, E_k(\cdot)/D_k(\cdot)\}$  and sends smart card to the user via a secure channel.
- v. After receiving the smart card, the user enters  $r_i$  into his/her smart card and finally the smart card contains  $\{G, E_i, H(\cdot), h(\cdot), S_i, T_i, E_k(\cdot)/D_k(\cdot), r_i\}$ .

### 2.2 Login Phase

The login phase is invoked when the user wants to login to the remote server. The following steps are performed:

- i. The user inserts his/her smart card into smart card reader and inputs the personal biometric  $F_i$  on the specific device to verify the user's biometric.
- ii. Verifying  $T_i = H(F_i)$ .
- iii. If the above condition does not hold, it means that the user  $U_i$  does not pass the correct biometric verification and the phase is terminated. If it is holds, the user passes the correct biometric verification and inputs his/her identity  $ID_i$  and password  $PW_i$  to perform the following operation.
- iv. After receiving the user's identity  $ID_i$  and password  $PW_i$ , the smart card computes the following:  $RPW_i^* = h(PW_i || r_i)$ ,  $D_i^* = h(ID_i || RPW_i^*) \cdot G$ ,  $C_i^* = E_i - D_i^*$ ,  $S_i^* = C_i + h(RPW_i^*) \cdot G$  and checks the condition whether the computed  $S_i^*$  matches with stored  $S_i$ . If it is same, then it implies that the user entered correct  $\{ID_i, PW_i\}$  and allows him/her to move for the next steps. Otherwise, it will terminate the session.
- v. The smart card generates a random nonce  $N_c$  and computes the following:  $AID_i = ID_i \oplus h(N_c)$ ,  $M_1 = h(N_c) \cdot G \cdot ID_i$  and  $M_2 = M_1 \cdot RPW_i$ ,  $M_3 = E_{C_x}(N_c, M_1)$  and  $M_4 = h(C_x || M_2 || N_c)$ , where  $C_x$  is the  $x$  coordinate of  $C_i = X \cdot G = (C_x, C_y)$ .
- vi. Then the user sends  $(AID_i, M_3, M_4)$  to the server over the public channel.

### 2.3 Authentication and Key Agreement Phase

This phase achieves mutual authentication and session key agreement between the user and the server after performing all the steps which are presented below.

- i. After receiving the login request message  $\{AID_i, M_3, M_4\}$ , the server first decrypts  $M_3$  using the key  $C_x$  and retrieves  $\{N_c, M_1\}$ . Then the server computes  $ID_i^* = AID_i \oplus h(N_c)$ ,  $M_1^* = h(N_c)G$ .  $ID_i^*$ ,  $M_2^* = M_1^* \cdot RPW_i$ ,  $M_4^* = h(C_x || M_2^* || N_c)$  and checks whether  $ID_i^* = ID_i$  and  $M_4^* = M_4$  or not. If this condition is hold then moves into the next step. Otherwise, terminates the session.
- ii. The server generates a random nonce  $N_s$  and computes the following:  $N_{sc} = N_s \oplus N_c$  and  $H_i = h(N_s || RPW_i || M_1)$ . Then server sends  $\{N_{sc}, H_i\}$  to the user.
- iii. After receiving the message  $\{N_{sc}, H_i\}$  from the server, the user first computes  $N_s = N_{sc} \oplus N_c$  using its own previously generated random nonce  $N_c$ . Then user checks  $H_i^* = h(N_s || RPW_i || M_1)$  using its own  $RPW_i$ ,  $M_1$  and previously generated random nonce  $N_c$ .
- iv. The user computes  $SK = h(N_s || N_c || RPW_i || M_2)$ ,  $Z_i = SK \cdot G + C_i$  and sends  $Z_i$  to the server for session key verification.
- v. The server computes  $SK = h(N_s || N_c || RPW_i || M_2)$ ,  $Z_i^* = SK \cdot G + C_i$  and compares  $Z_i^* = Z_i$ . If the comparison is matched it means that the session key verification holds correctly.

## 2.4 Password Change Phase

In this phase, whenever the user  $U_i$  wants to change his/her password, then he/she inserts his/her smart card into smart card reader and submits identity  $ID_i$ , password  $PW_i$ , and biometric information  $F_i$  and subsequently the smart card reader performs the following steps:

- i. Verifying  $T_i = H(F_i)$ .
- ii. If the above condition does not hold, it means that the user  $U_i$  does not pass the correct biometric verification and the phase is terminated. If it holds, the user passes the correct biometric verification and performs the next steps.
- iii. The user enters his/her identity  $ID_i$  and password  $PW_i$ , then the smart card computes the following:  $RPW_i^* = h(PW_i || r_i)$ ,  $D_i^* = h(ID_i || RPW_i^*) \cdot G$ ,  $C_i^* = E_i - D_i^*$ ,  $S_i^* = C_i^* + h(RPW_i^*) \cdot G$  and checks the condition whether the computed  $S_i^*$  is matched with stored  $S_i$ . If the comparison holds, it implies that the user has entered the correct  $\{ID_i, PW_i\}$  and allows him/her to move into the next steps. Otherwise, it will terminate the session.
- iv. The user inputs a new password  $PW_i^{new}$ , then the smart card computes:  $RPW_i^{new} = h(PW_i^{new} || r_i)$ ,  $D_i^{new} = h(ID_i || RPW_i^{new}) \cdot G$ ,  $E_i^{new} = E_i - h(ID_i || RPW_i) \cdot G + h(ID_i || RPW_i^{new}) \cdot G$ ,  $S_i^{new} = E_i - h(ID_i || RPW_i) \cdot G + h(RPW_i^{new}) \cdot G$ . Finally, the smart card replaces  $S_i$  with  $S_i^{new}$  and  $E_i$  with  $E_i^{new}$  into memory of the smart card and keeps rest of the smart card parameters unchanged.

## 3 Security Analysis Based on BAN Logic

This section verifies the validity of the presented scheme through Burrows-Abadi-Needham (BAN) logic [20]. The notation and logical postulates used in BAN logic is illustrated in Table 1. The BAN logic is a set of rules which can be used to verify whether information exchanged scheme is trustworthy and secured against various kinds of malicious attacks. To implement the BAN logic, the following steps are performed:

**Step 1:** In the BAN logic, the goals of our scheme can be presented as follows:

$$\text{Goal 1: } U_i \models (U_i \stackrel{SK}{\leftrightarrow} S)$$

$$\text{Goal 2: } U_i \models S \models (U_i \stackrel{SK}{\leftrightarrow} S)$$

$$\text{Goal 3: } S \models (U_i \stackrel{SK}{\leftrightarrow} S)$$

$$\text{Goal 4: } S \models U_i \models (U_i \stackrel{SK}{\leftrightarrow} S)$$

**Table 1** Notation of BAN logic

$\frac{P \models P \stackrel{K}{\leftrightarrow} Q, P \triangleleft \{X\}_K}{P \models Q \mid \sim X}$	<b>Message Meaning Rule:</b> If A believes that K is shared by P and Q and sees X encrypted with K, then P believes that Q once said X
$\frac{P \models Q \Rightarrow X, P \models Q \mid \sim X}{P \models X}$	<b>Jurisdiction Rule:</b> If P believes that Q has jurisdiction over X and P believes Q on the truth of X, then P believes on X
$\frac{P \models \#X}{P \models \#(X, Y)}$	<b>Freshness Conjunction Rule:</b> If P believes that freshness of X, then P believes that freshness of (X, Y)
$\frac{P \models \#X, P \models Q \mid \sim X}{P \models Q \mid \sim X}$	<b>Nonce Verification Rule:</b> If P believes that freshness of X and P believes that Q once said X, then P believes that Q believes X
$\frac{P \models \#X, P \models Q \mid \sim X}{P \models P \stackrel{K}{\leftrightarrow} Q}$	<b>Session Key Rule:</b> If P believes that X is fresh and P believes that Q believes X, which is the necessary parameters of session key. Then P believes that share the session key with Q

**Step 2:** We transform our presented scheme to the idealized form as follows:

Message 1.  $U \rightarrow S : \text{AID}_i, M_3: \{N_c, M_1\}_{C_x}, M_4$

Message 2.  $S \rightarrow U : N_{sc}, H_i: \langle N_s, M_1 \rangle_{\text{RPW}_i}$

**Step 3:** We make the following assumption to analyze the presented scheme.

$$A_1: S \mid \equiv \#(N_c, N_s)$$

$$A_2: U_i \mid \equiv \#(N_c, N_s)$$

$$A_3: S \mid \equiv S \stackrel{C_x}{\leftrightarrow} U_i$$

$$A_4: U_i \mid \equiv U_i \stackrel{\text{RPW}_i}{\leftrightarrow} S$$

$$A_5: S \mid \equiv U_i \Rightarrow N_c$$

$$A_6: U_i \mid \equiv S \Rightarrow N_s$$

**Step 4:** The main proofs are described as follows:

According to Message 1, we obtain

$$S_1: S \triangleleft (\text{AID}_i, M_3: \{N_c, M_1\}_{C_x}. M_4)$$

According to the assumption  $A_3$ ,  $S_1$  and the message meaning rule to obtain

$$S_2: S| \equiv U_i| \sim \{N_c, M_1\}$$

According to assumption  $A_1$  and freshness conjunction rule to obtain

$$S_3: S| \equiv \# \{N_c, M_1\}$$

According to  $S_2$ ,  $S_3$  and nonce verification rule to obtain

$$S_4: S| \equiv U_i| \equiv \{N_c, M_1\},$$

where  $N_c$  is the necessary parameter of the session key of the presented scheme.

According to the assumption  $A_5$ ,  $S_4$  and jurisdiction rule to obtain

$$S_5: S| \equiv \{N_c, M_1\}$$

According to the assumption  $A_1$ ,  $S_4$  and session key rule to obtain

$$S_6: S| \equiv \left( U_i \stackrel{\text{SK}}{\leftrightarrow} S \right) \quad \textbf{Goal 3 achieved.}$$

According to the assumption  $A_1$ ,  $S_6$  and nonce verification rule to obtain

$$S_7: S| \equiv U_i| \equiv \left( U_i \stackrel{\text{SK}}{\leftrightarrow} S \right) \quad \textbf{Goal 4 achieved.}$$

According to Message 2, we could obtain

$$S_8: U \triangleleft (N_{sc}, H_i: \langle N_S, M_1 \rangle_{RPW_i})$$

According to assumption  $A_4$ ,  $S_8$  and message meaning rule to obtain

$$S_9: U_i| \equiv S| \sim \{N_S, M_1\}$$

According to assumption  $A_2$  and freshness conjunction rule to obtain

$$S_{10}: U_i| \equiv \# \{N_S, M_1\}$$

According to  $S_9$ ,  $S_{10}$  and nonce verification rule to obtain

$$S_{11}: U_i| \equiv S| \equiv \{N_S, M_1\},$$

where  $N_s$  is the necessary parameter of the session key of the presented scheme.

According to assumption  $A_6$ ,  $S_{11}$  and jurisdiction rule to obtain

$$S_{12}: U_i | \equiv \{N_S, M_1\},$$

According to assumption  $A_2$ ,  $S_{11}$  and session key rule to obtain

$$S_{13}: U_i | \equiv \left( U_i \stackrel{SK}{\leftrightarrow} S \right) \quad \textbf{Goal 1 achieved.}$$

According to the assumption  $A_2$ ,  $S_{13}$  and nonce verification rule to obtain

$$S_{15}: U_i | \equiv S | \equiv \left( U_i \stackrel{SK}{\leftrightarrow} S \right) \quad \textbf{Goal 2 achieved.}$$

In this paper, the presented scheme has achieved the preferred goals as shown by the BAN logic. So this formal proof of the presented scheme is suitable to provide mutual authentication and session key agreement between participant entities securely.

## 4 Informal Security Analysis

In this section, we have further analyzed the presented scheme in the context of informal security. The presented security analysis demonstrates the effectiveness of such kind of authentication scheme in terms of security.

**Proposition 1** An outsider attacker cannot extract user's password  $PW_i$ , identity  $ID_i$ , and server's secret key  $X$  from the smart card parameters  $\{E_i, S_i, T_i, r_i\}$ , login request message  $\{AID_i, M_3, M_4\}$ , and reply message  $\{N_{sc}, H_i\}$  between the user and server.

*Proof* An outsider attacker obtains the user's smart card and extract secret information  $\{E_i, S_i, T_i, r_i\}$  from the smart card by some means, and he also intercepts the login request message  $\{AID_i, M_3, M_4\}$  and reply message  $\{N_{sc}, H_i\}$  between the user and server. But an attacker cannot extract the identity  $ID_i$ , password  $PW_i$ , and secret key  $X$  as follows:

1. From  $S_i = C_i + h(RPW_i) \cdot G = X \cdot G + h(PW_i || r_i) \cdot G$ , given  $r_i$ , an attacker has to guess server's secret key  $X$  and user's password  $PW_i$  at the same time to solve the mentioned equation. It is not computationally feasible in polynomial time to solve two unknown parameters password  $PW_i$  and secret key  $X$  from one equation.
2.  $E_i = C_i + D_i = X \cdot G + h(ID_i || RPW_i) \cdot G = X \cdot G + h(ID_i || h(PW_i || r_i)) \cdot G$ . In this case, the attacker has no knowledge of server's secret key  $X$ , user's identity  $ID_i$ , and password  $PW_i$ . Therefore, an attacker guesses three unknown values at the same time to solve the above equation which is not feasible in polynomial time.



3.  $AID_i = ID_i \oplus h(N_c)$ , an attacker cannot extract user's identity  $ID_i$  because it is computationally hard due to non invertible hash function.
4.  $M_3 = E_{C_x}(N_c, M_1)$ , where  $C_x$  is the x coordinate of  $C_i = X \cdot G = (C_x, C_y)$ . It is hard to compute secret key  $X$  due to elliptic curve discrete logarithm problem (ECDLP).
5.  $M_4 = h(C_x || M_2 || N_c) = h(X \cdot G || h(N_c) \cdot G \cdot ID_i \cdot h(PW_i || r_i) || N_c)$ . An attacker cannot extract the identity  $ID_i$ , password  $PW_i$  and secret key  $X$  due to elliptic curve discrete logarithm problem (ECDLP) and non-invertible hash function.
6.  $H_i = h(N_s || RPW_i || M_1) = h(N_s || h(PW_i || r_i) || h(N_c) \cdot G \cdot ID_i)$ , An attacker has to guess three unknown parameters  $N_s$ ,  $N_c$  and  $ID_i$  at the same time to extract the user's password from the equation  $H_i = h(N_s || RPW_i || M_1)$  which is computationally infeasible in polynomial time.

**Proposition 2** An insider attacker cannot extract server's secret key  $X$  from his/her own smart card parameters  $\{E_i, S_i, T_i, r_i\}$ , login request message  $\{AID_i, M_3, M_4\}$ , and reply message  $\{N_{sc}, H_i\}$  between the user and server.

*Proof* In this attack model, a legal but malicious user tries to extract the private key  $X$  of the sever by using his own identity, password, smart card parameters  $\{E_i, S_i, T_i, r_i\}$ , login request message  $\{AID_i, M_3, M_4\}$ , and reply message  $\{N_{sc}, H_i\}$ . In the following, we show that the malicious user cannot get the secret key  $X$ .

1. From  $S_i = C_i + h(RPW_i) \cdot G = X \cdot G + h(PW_i || r_i) \cdot G$ , given  $r_i$ . It is hard to compute secret key  $X$  due to the elliptic curve discrete logarithm problem (ECDLP).
2. Similarly, the attacker cannot extract secret key  $X$  from  $\{E_i, M_3, M_4\}$  parameters due to the same reason.

## 4.1 User Un-Traceability Attack

It is our assumption that an adversary has trapped two login request message  $\{AID_i, M_3, M_4\}$  and  $\{AID'_i, M'_3, M'_4\}$  during the execution of protocol and try to trace the both message are belonging to same user or not, where  $AID_i = ID_i \oplus h(N_c)$ ,  $M_3 = E_{C_x}(N_c, M_1)$  and  $M_4 = h(C_x || M_2 || N_c)$ . It may be noted that the each parameters  $\{AID_i, M_3, M_4\}$  are dependent on the random nonce  $N_c$ . Since the random nonce are distinct in each authentication session and valid for that session only. So in the presented scheme, the login request message  $\{AID_i, M_3, M_4\}$  and  $\{AID'_i, M'_3, M'_4\}$  are dissimilar in each authentication session. Therefore, an adversary cannot trace the user after intercepting the login message.

## 4.2 Privileged Insider Attack

Today, most of the authentication protocols are not secure due to the privileged insider attack. So it is very important to keep user's confidential information secret from the server. If a malicious administrator obtains the user's password by some means then he/she may use that password for accessing the other application servers where the user must registered himself/herself to every application server using the same identity  $ID_i$  and password  $PW_i$ . During the registration phase of the presented scheme, we provide  $RPW_i = h(PW_i || r_i)$  instead of plaintext password  $PW_i$  to the server, where  $r_i$  is a random number. So the insider attacker cannot extract  $PW_i$  from  $RPW_i$  due to the non-invertible hash function.

## 4.3 User-Server Impersonation Attack

In this attack model, we assume that the attacker intercepts the login request message  $\{AID_i, M_3, M_4\}$  and reply message  $\{N_{sc}, H_i\}$  and tries to impersonate as a legal user or server. However, *the Proposition 1* shows that an attacker cannot extract user's password  $PW_i$ , identity  $ID_i$  and server's secret key  $X$ . Thus, an adversary cannot computes valid login request  $\{AID_i, M_3, M_4\}$  and reply message  $\{N_{sc}, H_i\}$  without knowing user's password  $PW_i$ , identity  $ID_i$  and server's secret key  $X$ . Therefore, an attacker fails to impersonate as legitimacy entity of the presented scheme.

## 4.4 Password and Identity Guessing Attack

In the presented scheme, we have assumed that each user uses very low entropy identity  $ID_i$  and password  $PW_i$  which is easily guessable in polynomial time. However, *the Proposition 1* shows that an attacker cannot extract the user's identity  $ID_i$  and password  $PW_i$  from the smart card parameters  $\{E_i, S_i, T_i, r_i\}$  and also from the communicated messages  $\{AID_i, M_3, M_4, N_{sc}, H_i\}$  between the user and the server. Therefore, the presented scheme is secure against password and identity guessing attack.

## 4.5 Replay Attack

In the replay attack, the attackers intercepted the previous login message and later on transmit the same message to the server and try to impersonate as the legitimate entity. Suppose an attacker sends the previous intercepted message

$\{AID_i, M_3, M_4\}$  to the server, after receiving the message the server matches the received message with stored message. If it matches, then server rejects the attacker's login request. The presented protocol is secure against replay attack due to the random nonce  $N_c$  and  $N_s$  which are generated by user and server, respectively. Random nonce confirms that each login message is distinct in each session and valid for that session only.

#### 4.6 Stolen Smart Card Attack

To access the remote server, an attacker computes the login message  $\{AID_i, M_3, M_4\}$  by using the extracting parameters  $\{E_i, S_i, T_i, r_i\}$  from the stolen smart card. However, an attacker cannot compute valid login message  $\{AID_i, M_3, M_4\}$  as follows:

1.  $AID_i = ID_i \oplus h(N_c)$ , an attacker requires identity  $ID_i$  for computing  $AID_i$ . *The Proposition 1* shows that the attacker cannot extract identity  $ID_i$  from the smart card parameters  $\{E_i, S_i, T_i, r_i\}$ , login request message  $\{AID_i, M_3, M_4\}$ , and reply message  $\{N_{sc}, H_i\}$ . So, an attacker cannot compute  $AID_i$  without the knowledge of identity  $ID_i$ .
2.  $M_3 = E_{Cx}(N_c, M_1)$ , where  $Cx$  is the  $x$  coordinate of  $C_i = X \cdot G = (Cx, Cy)$  and  $M_1 = h(N_c) \cdot G \cdot ID_i$ . An attacker requires identity  $ID_i$  and secret key  $X$  for calculating the message  $M_3$ . *The Proposition 1* shows that the attacker cannot extract  $ID_i$  and  $X$  from the smart card parameters  $\{E_i, S_i, T_i, r_i\}$ , login request message  $\{AID_i, M_3, M_4\}$ , and reply message  $\{N_{sc}, H_i\}$ . So an attacker cannot compute  $M_3$  without the knowledge of identity  $ID_i$  and secret key  $X$ .
3. Similarly, an attacker cannot compute  $M_4 = h(Cx || M_2 || N_c)$  without the knowledge of  $ID_i$  and  $X$ .

Therefore, we may conclude that the presented scheme is secure against stolen smart card attack.

#### 4.7 Efficient Login and Password Change Phase

During the login and password change procedure of the presented scheme, the smart card reader detects the error very quickly if an attacker inputs the wrong information such as biometric template  $F_i$ , identity  $ID_i$ , and password  $PW_i$  to the card reader. As the result, an attacker cannot generate a fake login message which reduces extra computation and communication overhead as well as network congestion. Thus, the presented scheme provides efficient login and password change phase.

#### 4.8 Session Key Recovery Attack

In the presented scheme, the security of the session key  $SK = h(N_S || N_c || RPW_i || M_2)$  based on the non-invertible hash function. Moreover, the session key depends on the password  $PW_i$ , identity  $ID_i$ , random nonce  $N_S$  and  $N_c$ , which is generated by server and user, respectively. However, the *Proposition 1* shows that the attacker cannot extract user's password  $PW_i$  and identity  $ID_i$  from the smart card parameters  $\{E_i, S_i, T_i, r_i\}$  and communicated messages  $\{AID_i, M_3, M_4, N_{sc}, H_i\}$  between the user and the server. Without the knowledge of password  $PW_i$  and identity  $ID_i$ , an attacker cannot compute the session key  $SK$ . Thus, the presented scheme is secure against session key recovery attack.

#### 4.9 Perfect Forward Secrecy Attack

The perfect forward secrecy means, if the system's confidential information is compromised, then the secrecy of previous established session key should not be affected. In the presented scheme, we assume that user's password  $PW_i$  and identity  $ID_i$  are compromised by the attacker. Yet, the adversary cannot compute the previous session key  $SK = h(N_S, N_c, RPW_i, M_2)$  without the knowledge of random nonce  $N_c$  and  $N_S$ , which are generated by user and server, respectively. Therefore, the presented scheme is secure against perfect forward secrecy attack.

#### 4.10 Session Key Verification and Agreement

In this protocol, the user and server agreed upon a common session key  $SK = h(N_S || N_c || RPW_i || M_2)$  and verifies it using the following condition  $Z_i = SK \cdot G + C_i$ . To compute  $Z_i$ , an attacker has to know  $SK = h(N_S || N_c || RPW_i || M_2)$ ,  $C_i = X \cdot G$  and computes  $SK = h(N_S || N_c || RPW_i || M_2)$ , user's identity  $ID_i$  and password  $PW_i$  are needed and to compute  $C_i = X \cdot G$  server's secret key  $X$  is needed. The *Proposition 1* shows that an adversary cannot extract user's password  $PW_i$ , identity  $ID_i$  and server's secret key  $X$  from the smart card parameters  $\{E_i, S_i, T_i, r_i\}$  and communicated messages  $\{AID_i, M_3, M_4, N_{sc}, H_i\}$  between the user and the server. Since the only authorized entities can compute  $SK$  and  $Z_i$ . This demonstrates that the user and server can correctly verify the established session key.

## 5 Performance Analysis

In this section, we have compared the security and performance of the presented scheme with other existing relevant schemes [12–16]. In Table 2, we have summarized the communication cost, computation cost, and memory storage cost of the presented scheme and other relevant schemes. To analyze the computational complexity, we define the notation  $T_H$  is the one-way secure hash function,  $T_S$  is the symmetric key encryption/decryption operation, and  $T_{PM}$  is the point multiplication operation on elliptic curve cryptosystem. The presented protocol requires  $4T_H + 3T_{PM}$ ,  $12T_H + 8T_{PM} + 2T_S$ , and  $7T_H + 4T_{PM}$  operation for registration phase, login and authentication phase, and password change phase, respectively. In Table 3 and Fig. 1, we have provided communication cost and smart card storage cost comparison of the presented scheme with other relevant scheme [12–16]. In order to measure the communication cost, we have assumed that the length of identity  $ID_i$ , password  $PW_i$ , random nonce, elliptic curve point, and hash function  $h(\cdot)$  are all 160 bits length. Moreover, symmetric key encryption/decryption takes 512 bits. It is noticeable from Table 2 that the presented scheme achieves comparatively better communication cost than other schemes [13, 16]. In Table 3, we compared the presented scheme with existing related schemes in the context of different security functionalities. It is noticeable that the presented protocol is secure against relevant security attacks and achieves several security attributes than other schemes.

## 6 Conclusion

In this paper, we have presented a secure biometric based remote user authentication scheme using elliptic curve cryptosystem. The security of the presented scheme validated through both formal and informal way. The formal security analysis using BAN logic, which confirms that, the presented scheme achieves mutual authentication and session key agreement securely. The informal security analysis ensures that the presented scheme can resist various kinds of malicious attacks. The performance comparison demonstrates that the presented scheme is more secure and efficient than other relevant schemes. Moreover, the presented scheme can update the password on the user's demand without contacting the server.

**Table 2** Performance comparison: memory space, communication cost, and computational cost

TCC	$19T_H + 4T_{PM} + 11T_S$	$11T_H + 8T_{PM} + 8T_S$	$18T_H + 6T_{PM}$	$25T_H + 4T_{PM} + 4T_S$	$30T_H + 12T_{PM}$	$23T_H + 15T_{PM} + 2T_S$
CC	$160 * 3 + 512 * 2 = 1504$	$160 * 5 + 512 = 1312$	$160 * 7 = 1120$	$160 * 4 + 512 * 2 = 1664$	$160 * 7 = 1120$	$160 * 5 + 512 = 1312$

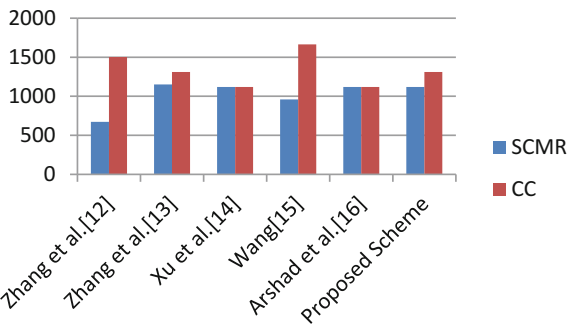
Note: *SCMR* smart card memory requirement; *CCRP* computation cost of registration phase; *CCLA* computation cost of login and authentication phase; *CCPH* computation cost of password change phase; *TCC* total computation cost; *CC* communication cost

**Table 3** Security feature comparisons among the presented scheme and other relevant scheme

Security requirement	Zhang et al. [12]	Zhang et al. [13]	Xu et al. [14]	Wang [15]	Arshad and Nikooghadam [16]	Presented scheme
A <sub>1</sub>	Yes	No	Yes	No	Yes	Yes
A <sub>2</sub>	Yes	No	Yes	No	Yes	Yes
A <sub>3</sub>	Yes	No	Yes	Yes	Yes	Yes
A <sub>4</sub>	Yes	No	Yes	Yes	Yes	Yes
A <sub>5</sub>	No	No	Yes	Yes	Yes	Yes
A <sub>6</sub>	No	Yes	Yes	Yes	Yes	Yes
A <sub>7</sub>	No	Yes	Yes	Yes	Yes	Yes
A <sub>8</sub>	Yes	Yes	No	Yes	No	Yes
A <sub>9</sub>	No	Yes	No	Yes	No	Yes

Note: A<sub>1</sub> resist password guessing attack, A<sub>2</sub> resist identity guessing attack, A<sub>3</sub> resist impersonation attack, A<sub>4</sub> resist privileged insider attack, A<sub>5</sub> resist replay attack, A<sub>6</sub> resist user un-traceability attack, A<sub>7</sub> resist forward secrecy attack, A<sub>8</sub> session key verification, A<sub>9</sub> efficient login and password change phase

**Fig. 1** Communication and storage overhead (bits) of different authentication scheme. Note: *SCMR* smart card memory requirement, *CC* communication cost



References

1. L. Lamport, Password authentication with insecure communication. Commun. ACM **24**(11), 770–772 (1981)
2. C.-C. Lee, L.-H. Li, M.-S. Hwang, A remote user authentication scheme using hash functions. ACM SIGOPS Oper. Syst. Rev. **36**(4), 23–29 (2002)
3. M. Peyravian, C. Jeffries, Secure remote user access over insecure networks. Comput. Commun. **29**(5), 660–667 (2006)
4. X.-M. Wang et al., Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards. Comput. Stand. Interfaces **29**(5), 507–512 (2007)
5. S. Kumari, M.K. Khan, X. Li, An improved remote user authentication scheme with key agreement. Comput. Electr. Eng. **40**(6), 1997–2012 (2014)
6. C.T. Li, M.S. Hwang, An efficient biometrics-based remote user authentication scheme using smart cards. J. Netw. Comput. Appl. **33**(1), 1–5 (2010)
7. C.H. Lin, Y.Y. Lai, A flexible biometrics remote user authentication scheme. Comput. Stand. Interfaces **27**(1), 19–23 (2004)

8. B.T. Nathan, R. Meenakumari, S. Usha, *Formation of Elliptic Curve Using Finger Print for Network Security*. In Process Automation, Control and Computing (PACC), 2011 International Conference on IEEE, pp. 1–5
9. X. Li, J.W. Niu, J. Ma, W.D. Wang, C.L. Liu, Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. *J. Netw. Comput. Appl.* **34**(1), 73–79 (2011)
10. U. Subramaniam, K. Subbaraya, A biometric based secure session key agreement using modified elliptic curve cryptography. *Int. Arab J. Inf. Technol. (IAJIT)* **12**(2) (2015)
11. C.-T. Li, A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card. *IET Inf. Secur.* **7**(1), 3–10 (2013)
12. L. Zhang et al., Two-factor remote authentication protocol with user anonymity based on elliptic curve cryptography. *Wireless Pers. Commun.* **81**(1), 53–75 (2015)
13. Y. Zhang et al., An efficient password authentication scheme using smart card based on elliptic curve cryptography. *Inf. Technol. Control* **43**(4), 390–401 (2014)
14. X. Xu, P. Zhu, Q. Wen, Z. Jin, H. Zhang, L. He, A secure and efficient authentication and key agreement scheme based on ECC for telecare medicine information systems. *J. Med. Syst.* **38**(6) (2014)
15. L. Wang, Analysis and enhancement of a password authentication and update scheme based on elliptic curve cryptography. *J. Appl. Math.* (2014)
16. H. Arshad, M. Nikooghadam, Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems. *J. Med. Syst.* **38**(12) (2014)
17. Z. Tan, A user anonymity preserving three-factor authentication scheme for telecare medicine information systems. *J. Med. Syst.* **38**(3), 1–9 (2014)
18. Y. Lu et al., An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem. *J. Med. Syst.* **39**(3), 1–8 (2015)
19. H.L. Yeh et al., Robust elliptic curve cryptography-based three factor user authentication providing privacy of biometric data. *IET Inf. Secur.* **7**(3), 247–252 (2013)
20. M. Burrows, M. Abadi, R. Needham, A logic of authentication. *ACM Trans. Comput. Syst.* **8**(1), 1836 (1990)



Proceedings of the International Conference on  
Microelectronics, Computing & Communication Systems  
MCCS 2015

Nath, V. (Ed.)

2018, XV, 388 p. 241 illus., 174 illus. in color.,

Hardcover

ISBN: 978-981-10-5564-5