Journal of Information Security and Applications 51 (2020) 102443

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

ESEAP: ECC based secure and efficient mutual authentication protocol using smart card

Adesh Kumari[a], Srinivas Jangirala[b,*], M. Yahya Abbasi[a], Vinod Kumar[c], Mansaf Alam[d]

[a] Department of Mathematics, Jamia Millia Islamia, New Delhi 110025, India
[b] Jindal Global Business School, O. P. Jindal Global University, Haryana 131001, India
[c] Department of Mathematics, PGDAV College, University of Delhi, New Delhi 110065, India
[d] Department of Computer Science, Jamia Millia Islamia, New Delhi 110025, India

ARTICLE INFO

Article history:

Keywords:
Mutual authentication
Elliptic curve cryptography
Smart card
Random oracle model
Security and privacy

ABSTRACT

Smart card based user server mutual authentication framework is famous for safe communication via unfavorable and insecure communication system. The authenticated user and server communicate to each other and share information via Internet. Recently, Wang et al. suggested a lightweight password-assisted two factor authentication framework using smart card. We reviewed their scheme and observed that it does maintain security and privacy off-line password guessing attack and also impersonation attack. We proposed enhance elliptic curve cryptography(ECC) based authentication framework for the same environment. The proposed scheme ESEAP is secure resilience of many attractive security attributes and features like off-line password guessing attack, no password verifier-table, smart card loss attack, anonymity, mutual authentication, replay attack, impersonation attack, server spooling attack, no clock-synchronization attack, forward secrecy, insider attack, message authentication, provision of key agreement, parallel attack, sound repairability, no password exposure, timely typo detection, resistance to know attacks, password friendly, user unlinkability and server unlinkability. Further, the paper shows formal security analysis of the ESEAP which based on random oracle model. We compared the presented protocol with other related protocols in the same environment, and show that ESEAP is more efficient in terms of computation and communication cost. As a result, the presented protocol can be utilized over public communication channel.

1. Introduction

With the speedy progress of Internet, online works became an imperative factor of public activity. However, due to the accessibility of the communication network, people have to face the safety risks when they savor the assistance of the network. Literally, it is not astonishing to a collection of dispatch of client message exposure instantly. The authentication is a powerful and effective method to the clients to appreciates resources in different accessible environment like telecare medical information system, cloud computing, E-health, wireless body area network and wireless sensor [1,2]. The history of password-assisted authentication protocol tracked backward in 1981, when Lamport [3] essentially encouraged an authentication framework. Then, authentication protocol equipped with passwords were extensively endorsed and several authentication protocol appeared [4,5]. Nonetheless, these protocols have an instinct weakness: a password index must be controlled by a system server, which has two issues: (1) Most of protocol are susceptible to stolen-verifier attack and (2) to maintain and secure a password index at very high cost. To overcome these difficulties, Chang and Wu presented password based two-factor authentication protocol [6]. In such work, the clients are not only intimated to find the specific password but also by using smart card, clients can approach to area by collaborating with network server.

The smart card established authentication protocol with two partners: user and server. Moreover, it consists of four basic phases. 1). In registration phase, client submits his/her identity and password to server and server provides it to smart card via secure channel. 2). In login phase, users address a new entry request to server. 3). In verification phase, user and server authenticate each other. If, they validate each other, then authentication mechanism properly completed, the user can relish the support from the server side. In last phase, user change password phase where client wants to replace his/her password in proper

* Corresponding author.
E-mail addresses: adeshbhucker@gmail.com (A. Kumari), sjangirala@jgu.edu.in (S. Jangirala), mabbasi@jmi.ac.in (M.Y. Abbasi), vinod.iitkgp13@gmail.com (V. Kumar), malam2@jmi.ac.in (M. Alam).

https://doi.org/10.1016/j.jisa.2019.102443
2214-2126/

manner. In briefly, a strong smart card assisted authentication framework should establish only when the client who is not only agree to the specific password but also his/her smart card entry can be permitted. Chang and Wu, presented famous two factor authentication protocol and discussed some famous information [7,8]. Compared with the preceding two- factor password-assisted frameworks, these protocols are less of cost and contains better cryptographic capability in communication environment. Till now, smart card-based authentication framework becomes convenient to protect the security and privacy of communication network in authentication protocols. In this article, we proposed an ECC-assisted authentication framework which is secure and efficient framework over insecure communication network.

1.1. Related work

In 2005, Fan et al., suggested two factor authentication scheme that declines to attain client session key and anonymity installation [9]. As, it is delivered on Rabin's public key cryptosystem, Fan et al., is low adequate when it is compared with new outcomes based on ECC. To maintain client security and privacy, Das et al. [10] presented identity-assisted password authentication protocol. In these years, it became an important field for research drawing of authenticated framework using dynamic identity [11–17]. Juang et al., suggested an anonymous authenticated scheme [11]. Sun et al. [13] and Li et al. [14] shown that Juang et al., protocol fails against untraceability, anonymity off-line attack and lost smart card attack, and password change property. In 2014, Huang et al. proposed [18] smart-card-based password-authenticated key agreement in distributed systems. They claimed that protocols [19,20] failed against offline-attack and online attack with smart card. In same year, Wang and Wang presented protocol [21] on the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions. In this protocol Wang and Wang reviewed schemes [22–24] and discussed the lack of anonymity properties in theses framework. In 2015, Wang et al. presented [25] an anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment. In the proposed work Wang et al. have shown the analysis of protocols [26,27] and discussed demerits of these schemes. In 2016, Wang et al. proposed framework [28] The request for better measurement: A comparative evaluation of two-factor authentication schemes. In this work, Wang et al. provided security drawback of Li et al. [29], Kumari and Khan [30], Odelu et al. [31] and Fahad and Muhaya [32] protocols. According as [28], Li et al. [29] protocol is not secure against offiline/online guessing attack, Kumari and Khan [30] protocol fails against de-synchronization attack, stolen/piced up attack, guessing attack and no forward secrecy, Odelu et al. [31] protocol is not manages smart card loss attack, offline guessing attack and flaws in formal security proofs and, Fahad and Muhaya [32] protocol fails against user impersonation attack, smart card loss attack and no forward secrecy. In 2018, Luo et al. [33] presented lightweight three factor scheme for real-time data access in wireless sensor networks which is fail against smart card loss attack, user anonymity, reply attack, forward secrecy, no password verifier table, sound repairability, resistance to know attack and user and server unlinkability. Ma et al. [34], Madhusudhan and Mittal [35] and Wang et al. [36] displayed that different newly scheduled identity-based two-factor authenticated schemes have one or more vulnerability, such as weakness against lost smart card attack, lack of forward secrecy, off-line guessing attack, untraceability and anonymity. It is observed that, in order to offer client anonymity, mostly all dynamic identity-assisted authentication schemes need a supplementary synchronization method to sustain the flexibility of the identity between the server and the user. However, this resilience can be cracked simply, and the client may no longer be capable to login the server [16]. Chaudhry et al. arranged two protocols [15,37] that are disclosed to resolve anonymity and several other beautiful qualities, but both of them does not reinforce smart card revocation, and the second protocol [37] does not offer password change method. Besides, Xie et al. found that the first protocol [15] declined to obtain forward secrecy although it alleged so, because its earlier session keys can be launched if the attacker gets approach to the client's password, smart card and scheme transcriptions of past sessions. Wang et al. [38] presented two birds with one stone: Two-factor authentication with security beyond conventional bound which fails against user and server unlinkability. Buyn et al. [32] proposed cryptanalysis and security enhancement of Zhu's authentication scheme for TMIS which fails smart card loss attack, user anonymity, no password verifier-table and no password exposure. Amin et al. [39] proposed an anonymity preserving and lightweight multi-medical server authentication protocol for TMIS which fails against smart card loss attack, user anonymity, mutual authentication and sound repairability, no password exposure, resistance to know attacks and off-line password guessing attack. Wang et al. [16] proposed preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity which fails against user anonymity, server spooling attack, no clock-synchronization attack and forward secrecy. Odelu et al. [31] proposed an effective and robust secure remote user authenticated key agreement scheme using smart cards in wireless communication systems which is not safe against user anonymity, parallel session attack and Password friendly method. Wu et al. [40] presented a new and secure authentication scheme for wireless sensor networks with formal proof which fails against smart card loss attack, user anonymity, sound repairability, password friendly and user and server unlinkability. Ali et al. [41] suggested a secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring. It fails against smart card loss attack, user anonymity, no clock-synchronization attack, Forward secrecy and no password exposure property. Luo et al. [33] proposed a lightweight three factor scheme for real-time data access in wireless sensor networks which fails in smart card loss attack, user anonymity, replay attack, no password verifier-table, sound repairability, forward secrecy resistance to know attacks, user and server unlinkability. Roy et al. [42] proposed provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications which fails against sound repairability, timely typo detection and resistance to know attacks. Recently, Wang et al. [43] presented a password-assisted authentication framework using smart card. We reviewed their framework and found the security weaknesses of this protocol like stolen-verifier attack, no protection for session key, off-line password guessing attack and it is unsecure against to impersonation attack.

1.2. Motivation and contribution

Even though VCC-assisted authentication protocols [16,31–33,38–51] have been presented over the last few decades, their achievement is not enough. Moreover, these frameworks disrupt the essential obligations of protocols, so resulting in elementary breach. Recently, Wang et al. [43] proposed protocol that can work in the communication system.

- It is analyzed and demonstrated as below:
  - Their protocol does not maintain safety against off-line password guessing attack.
  - Their framework does not safe against impersonation attack.
- In order to gain security against the aforementioned security features and attributes, an authentication scheme for smart

Table 1
Notations/Symbol used.

Symbol	Description	Symbol	Description
U	The user	S	The server
SC	The smart card	ID_i	The unique identity of i^{th} participant
l	The security parameter	q, p	Large primes
F_q	The prime finite field	$EC\ (F_q)$	Elliptic curve over F_q
G	Additive ECC group	K_i	Key generated by participant i
x	Private key of server	g	The base point of G
PW_i	Password of i^{th} entity	Z_q^*	Additive group of order q
$h\ (\cdot)$	Cryptographic secure one way secure hash function	$SK_{ij}(.)$	The session key between entities i and j
$\stackrel{?}{=}$	Whether i equals j	$\|$	Concatenation operation
$\oplus$	Bitwise XOR operation	$\mathcal{A}$	An adversary
RP	Registration phase	LAP	Login and authentication phase

card is proposed which is useful for communication environment. Our protocol has many significant characteristics which are explained as below:

- We proposed ECC based two factor mutual authentication for smart card.
- We introduce the "*honeywords*" as a *Honey_list* which is consistently concept of security system into cryptographic protocol.
- Mutual authentication is achieved between user and server via insecure public channel.
- Further, the presented scheme is secure against many security attacks, features and satisfies different attributes such as smart card loss attack, anonymity, mutual authentication, replay attack, impersonation attack, server spooling attack, no clock-synchronization attack, provision of key agreement, insider attack, message authentication, off-line password guessing attack, parallel attack, resistance to know attacks, no password verifier-table, sound repairability, no password exposure, forward secrecy, timely typo detection and password friendly.
- We show the correctness of the proposed protocol by the formal security analysis which is based random oracle model.
- We compare ESEAP with other related protocols and found that it requires minimum computation and communication cost.

1.3. Paper organization

The rest of the paper is mapped as. Section 2, We described the preliminaries. Section 3, We reviewed Wang et al.'s. protocol. Section 4, The cryptanalysis of Wang et al. protocol. Section 5, Formal security model. Section 6, The ESEAP authentication protocol. Section 7, Security analysis. Section 8, Performance analysis. Lastly, we have given a future direction and conclusion. Now, we start with Table 1 which contains symbol/notations used for the paper.

2. Preliminaries

2.1. Background of ECC

Let $EC(F_q)$ be denotes an elliptic curve over the prime finite field F_q, where q be the large prime number. An equation of elliptic curve over F_q is given by $v^2 = u^3 + \alpha u + \beta \ mod\ q$, where $\alpha, \beta \in F_q$. The elliptic curve is said to be non singular if $4\alpha^3 + 27\beta^2 \ mod\ q \neq 0$. The additive elliptic curve group G is defined as $G = \{(u, v) : u, v \in F_q;\ (u, v) \in \mathcal{E}\} \bigcup \{\Phi\}$, where the point Φ is known as asymptotic point which work as the identity element or zero element in G. Some operations on the group G are as follows [52]:

1. Let $\bigvee = (u, v) \in G$, then define $-\bigvee = (u, -v)$ and $\bigvee + (-\bigvee) = \Phi$.
2. If $\bigvee_1 = (u_1, v_1),\ \bigvee_2 = (u_2, v_2) \in G,$ then $\bigvee_1 + \bigvee_2 = (u_3, v_3)$, where $u_3 = \rho^2 - u_1 - u_2 \ \ mod\ q,\ \ v_3 = \rho(u_1 - u_2) - v_1 \ \ mod\ q$ and
$$\rho = \begin{cases} \frac{v_2 - v_1}{u_2 - u_1} \ mod\ q & if\ \ \bigvee_1 \neq \bigvee_2 \\ \frac{3u_1^2 + \alpha}{2v_1} \ mod\ q & if\ \ \bigvee_1 = \bigvee_2 \end{cases}$$
3. Let $\bigvee = (u, v) \in G$ then, scalar multiplication in G is defined as: $\eta.\bigvee = \bigvee + \bigvee + \bigvee \ldots\ldots\ldots + \bigvee\ (\eta - times)$.
4. If g is the generator of G with order η, then $\eta.g = \Phi$.

The key size comparison between ECC and RSA is discussed in Table 2:

Table 2
Key size comparison between ECC and RSA [53].

S. No.	ECC key size(bits)	RSA key size(bits)	Key size ratio
1.	163	1024	1:6
2.	256	3072	1:12
3.	384	7680	1:20
4.	512	15360	1:30

- **Elliptic curve discrete logarithms problem (ECDLP)**: For given $\bigvee_1, \bigvee_2 \in G$ find $\mu \in Z_q^*$ such that $\bigvee_2 = \mu.\bigvee_1$, which is hard.
- **Elliptic curve computational Diffie-Hellman problem (ECCDHP)**: For $\alpha, \beta \in Z_q^*$ and g is the generator of G, given (g, $\alpha.g$, $\beta.g$), then to compute $\alpha.\beta.g$ is hard in G.

2.2. Dolev-Yao (DY) threat model

In this paper, we consider the common Dolev-Yao model as mentioned in [54–56]. Where according to the capabilities of the adversary $\mathcal{A}$, following assumptions are made:

- $\mathcal{A}$ can access the public communication channel. He/she can retrieve, modify, replay, inject new message and can discard any communication information.
- $\mathcal{A}$ is presumed to be protected, therefore cannot obtain the secret key of participants.
- $\mathcal{A}$ knows the public identities of all the users and server.
- $\mathcal{A}$ can be an intruder or can be an insincere user/server of the underlying system.

3. Review of Wang et al. protocol

In this section, we reviewed the Wang et al. protocol [43] which is having the followings phases:

3.1. Registration phase

If U_i tries to approaches to S, he/she has to register to S and they have the following steps:

Step 1. U_i chooses ID_i and PW_i. Then, the system takes a arbitrary value b and executes $PWR_i = h(PW_i \| b)$. U_i sends message $\{PWR_i, ID_i\}$.

Step 2. After receiving $\{PWR_i, ID_i\}$ at T_{rg}, S executes $A_i = h(h(ID_i) \oplus h(PW_i \| b))\ mod\ n_0$, where n_0 is integer value such as $2^4 \leq n_0 \leq 2^8$, and verifies whether ID_i has been in the *User_List*. If it does not, S generates a fresh entry for U_i as $\{ID_i, y_i, T_{rg}, Honey\_List\}$, where y_i is a unique nonce to U_i selected by S, *Honey_List* is a index to data record the number of login losses and initialized to *NULL*; Otherwise, S revises T_{rg} and y_i in the *User_List*. Then S computes $K_i = h(ID_i \| x \| y_i \| T_{rg}), L_i = K_i \oplus PWR_i$. S sends SC with $\{A_i, L_i, n_0, p, g, y, h(.)\}$.

Step 3. On collecting SC, U_i inserts b into SC.

3.2. Login and authentication phase

In this phase, U_i and S take the followings steps:

Step 1. U_i enters SC and takes ID_i' and PW_i'. SC executes $PWR_i' = h(PW_i' \| b), A_i' = h(h(ID_i') \oplus h(PWR_i'))\ mod n_0)$, and checked $A_i' \stackrel{?}{=} A_i$. Further, SC takes a random value r_1 and calculates $C_1 = g^{r_1}\ mod p, C_2 = y^{r_1}\ mod\ p, D_i = ID_i \oplus h(C_1 \| C_2)$, and $K_i' = L_i \oplus PWR_i', M_1 = h(D_i \| C_1 \| C_2 \| K_i')$. U_i sends $\{C_1, C_2, M_1\}$ to S.

Step 2. On receiving $\{C_1, C_2, M_1\}$, S executes $C_2' = (C_1)^x\ mod\ p$ and $ID_i' = D_i \oplus h(C_1 \| C_2')$. Then, S obtains the *User_List* to obtain ID_i such that $ID_i \stackrel{?}{=} ID_i'$. If, it is not so, then S refuses the entry request and makes the *Honey_List* to be *Honey_List* + 1. Once the amount of exceeds the prearranged threshold, S judges SC has been cracked, thus appends SC till U_i re-registers. Otherwise, S determines y_i and T_{rg} from the *User_List* and executes $K_i = h(D_i \| x \| y_i \| T_{rg})$ and $M_i' = h(D_i \| C_1 \| C_2' \| K_i)$. Then, S checks U_i by verifying whether $M_1' \stackrel{?}{=} M_1$. If it does not true then, S terminates the session and sets *Honey_List* = *Honey_List* + 1, in that time the number of *Honey_List* exceeds the prearranged threshold, appends the SC till U_i re-registers; otherwise, S initializes the RSA algorithm and generates private (n, d) and public key (n, e), and computes $M_2 = h(D_i \| C_1 \| C_2' \| K_i \| n \| e)$. S sends $\{M_2, n, e\}$ to U_i.

Step 3. On receiving message, SC computes $M_2' = h(D_i \| C_1 \| C_2 \| K_i' \| n \| e)$ and $M_2' \stackrel{?}{=} M_2$. Then, S chooses an arbitrary value r_2, where $0 \leq r_2 \leq n$ and executes $C_3 = r_2^e\ mod\ n$ and $M_3 = h(D_i \| C_3 \| C_2 \| K_i' \| r_2 \| C_2)$. U_i sends $\{M_3, C_3\}$ to S.

Step 4. On receiving message, S executes $r_2' = C_3^d$ mod n and $M_3' = h(D_i \| C_3 \| C_2 \| K_i \| r_2' \| C_2')$ and verifies $M_3' \stackrel{?}{=} M_3$. Then, S and U_i executes session key as $SK = h(D_i \| C_1 \| C_2 \| C_3 \| K_i' \| r_2)$ and $SK = h(D_i \| C_1 \| C_2' \| C_3 \| K_i \| r_2')$, respectively.

3.3. Password change phase

When U_i wants to changes his/her password, he/she has to perform the following steps:

Step 1. U_i inserts SC and takes ID_i', PW_i' and fresh password PW_i^{new}. SC executes $PWR_i' = h(PW_i' \| b), A_i' = h(h(ID_i') \oplus h(PWR_i')\ mod\ n_0)$ and verified $A_i' \stackrel{?}{=} A_i$. Then, SC executes $PWR_i^{new} = h(PWR_i^{new} \| b), A_i^{new} = h(h(ID_i') \oplus h(PWR_i^{new})\ mod\ n_0)$ and $L_i^{new} = L_i \oplus PWR_i' \oplus PWR_i^{new}$.

Step 2. Changes L_i with L_i^{new} and A_i with A_i^{new}, respectively.

3.4. Revocation phase

If U_i lost or breached his/her SC, he/she takes the followings steps:

Step 1. U_i enters SC into a reader, takes ID_i' and PW_i'. SC executes $PWR_i' = h(PW_i' \| b), A_i' = h(h(ID_i') \oplus h(PWR_i')\ mod\ n_0)$, verified $A_i' \stackrel{?}{=} A_i$. Further, SC takes random value r_1, computes $C_1 = g^{r_1}\ mod\ p$, $C_2 = y^{r_1}\ mod\ p, D_i = ID_i \oplus h(C_1 \| C_2), K_i' = L_i \oplus PWR_i', M_1 = h(D_i \| C_1 \| C_2 \| K_i')$. U_i sends the $\{C_1, C_2, M_1, Revoke\_request\}$ to S.

Step 2. On receiving message, S executes $C_2' = (C_1)^x\ mod\ p, ID_i' = D_i \oplus h(C_1 \| C_2)$. Then, S finds the *User_List* for find the identity $ID_i \stackrel{?}{=} ID_i'$. If it does not hold S refuses the offers and selects *Honey_List* to be *Honey_List* + 1. If once the amount of *Honey_List* outstrips the fixed threshold, S realizes that the SC has been cracked, thus appends SC till U_i re-registers. Otherwise, S determines y_i and t_{rg} from the *User_List*, and executes $K_i = h(ID_i \| x \| y_i \| T_{rg}), M_1' = h(ID_i \| C_1 \| C_2 \| K_i')$ and check whether $M_1' \stackrel{?}{=} M_1$. Then, S selects the *Honey_List* to be *Honey_List* + 1. If once the amount of *Honey_List* outstrips the fixed doorsteps, appends the smart card till U_i re-registers. Otherwise, S sets y_i and *NULL*, so next time, U_i cannot login strongly.

3.5. Re-registration phase

In this phase, U_i takes the followings steps:

Step 1. U_i sends $\{PWR_i, D_i\}$ to S.

Step 2. Firstly, S verifies whether the identity ID_i is in the *Honey_List* + 1 and whether the account of the U_i is revoked or his/her SC is suspended. If so, S computes the registration as discussed in Section 3.1.

4. Cryptanalysis of Wang et al. protocol

In this section, we demonstrated the Wang et al.'s framework which does not manage security against off-line password guessing attack through a privileged-insider attack. The discussion is shown as below:

4.1. Off-line password guessing attack:

Step 1. Suppose a privileged-insider of the server S being an adversary $\mathcal{A}$ knows the registration information $\{PWR_i, ID_i\}$, where $PWR_i = h(PW_i \| b)$.

Step 2. Assume after registration phase, the adversary $\mathcal{A}$ has the lost/stolen smart card SC of the registered user. The adversary has the information $\{A_i, L_i, n_0, p, g, y, h(.)\}$ including random secret b.

Step 3. The adversary $\mathcal{A}$ can now guess a password PW' and checks if $PWR_i \stackrel{?}{=} h(PW' \| b)$ If it is so, the adversary $\mathcal{A}$ is successful in guessing the password.

In addition to this, the adversary $\mathcal{A}$ can also get $K_i = L_i \oplus PWR_i$ which is used in computing the session key and also the communicated messages. If identity of any user is revealed then the adversary $\mathcal{A}$ can easily impersonate the user by intercepting the communicated messages.

5. Formal security model

In this section, we discuss the random oracle model for the presented framework. It is established in [57–59] which is denoted by Γ. In Γ, only two entities are associated: 1) Legitimate user U and 2) Server S. Now, we take some assumption about the execution Γ and the stability of any probabilistic polynomial time bounded attacker E.

- Both U and S are permitted to compute in Γ time. We take $\prod^i$ be the i^{th} occurrence of the scheme party $\prod$ (either U or S). We define U^i/S^j in the i^{th}/j^{th} occurrence of U/S.
- Assume that three states for an oracle: accept, reject and $\perp$. If an oracle obtains a normal message, the accept state is achieved. If wrong message is obtained, the reject state is achieved. Otherwise, if no reply is achieved, $\perp$ appears.
- Assume that U gathers an identity ID_U, and a small -magnitude password PW_i. Note that PW_i is chosen randomly from the dictionary Υ. In RP, S and U both of them jointly build of injection conversion of the message $\{G, g, h(.), L, P_1, P_2, P_3\}$ in to SC of U.
- $\mathcal{A}$ and any participant occurrence $\prod^i$ (either S or U) collaborates by computing oracle queries, which allows the space of E to attack the semantics insurance of the session executed by the parties of Γ.
- All the information, which are measured by $\prod^i$ and its participant during the completing of Γ are interconnected over an apprehensive channel, which is entirely managed by $\mathcal{A}$, i.e., the attacker can block, inject, intercept, remove or modify, any information communicated through it [59].
- We consider some attack scopes of $\mathcal{A}$. $\mathcal{A}$ can either (a) steal U's SC or hold U' lost SC, and then implement different investigation as explained in [60–62] for accessing the secret message from it. After obtaining the obscure information. $\mathcal{A}$ implements some off-line method and get U's password. (b) get U's password precisely. It is to be acclaimed that, $\mathcal{A}$ cannot execute both (a) and (b) simultaneously [63].

In this model, $\mathcal{A}$ is permitted to implement the actual attack to crack the linguistic preservation of the session key executed by $\prod^i$ and it is associated in a session of Γ. We adopt all the queries and definitions based [64].

6. The ESEAP protocol

The enhanced scheme contains following phases:

6.1. Initialization phase

Initially, S chooses non singular elliptic curve $EC(F_q)$ of the equation $v^2 = u^3 + \alpha u + \beta$ over F_q, where $Z_q(q > 2^{160})$ is the prime finite field. S selects α, $\beta \in F_q$ and satisfy the condition $4\alpha^3 + 27\beta^2 \ mod \ q \neq 0$. S selects g be the generator of G. S generates random number x and sets as secret key.

6.2. Registration phase

In this phase, U finds registration as below:

Step R1. To register with S, U chooses ID_U, passwords PW_U and takes random value $a \in Z_q^\star$. Moreover, U computes $PWU = h(ID_U \| PW_U \| a)$ and sends message $\{PWU, ID_U\}$ to S.

Step R2. On receiving message, S generates random number $b \in Z_q^\star$ and sets $User\_List : \{ID_U, b, Honey\_List\}$. Further, S computes $c = ID_U \oplus b$, $B_1 = h(ID_U \| x \| c \| b)$ and $L_1 = B_1 \oplus PWU$. Furthermore, S generates smart card as $\{G, g, h(.), L_1, c\}$ and sends to U via secure channel.

Table 3
Registration Phase.

User/Smart card	Server
Chooses ID_U and PW_U	
$a \in Z_q^\star$	
$PWU = h(ID_U \| PW_U \| a)$	
$\{PWU, ID_U\}$	
. . . . . $\rightarrow$	$b \in Z_q^\star$
(via secure channel)	$User\_List : \{ID_U, b, Honey\_List\}$
	$c = ID_U \oplus b$
	$B_1 = h(ID_U \| x \| c \| b)$
	$L_1 = B_1 \oplus PWU$
	SC with $\{G, g, h(.), L_1, c\}$
	$\leftarrow$ . . . . .
$P_1 = a \oplus h(ID_U \| PW_U)$	(via secure channel)
$P_2 = h(PWU \| P_1)$	
$P_3 = a \oplus ID_U \oplus c$	
Delete c from SC and Inserts P_1, P_2, P_3 in SC	

Step R3. Upon receiving $\{G, g, h(.), L_1, c\}$, U executes $P_1 = a \oplus h(ID_U \| PW_U)$, $P_2 = h(PWU \| P_1)$ and $P_3 = a \oplus ID_U \oplus c$. Further, U delete c from SC and inserts P_1, P_2, P_3 in SC.

The process of registration phase is displayed in Table 3.

6.3. Login and authentication phase

U registers to S successfully, he/she sends the entry request to S. These are mutually authenticated to each other and perform the following steps :

Step LA1. U enters SC in to card reader, takes ID_U' and PW_U'.

Step LA2. SC generates random number $a^* = P_1 \oplus h(ID_U \| PW_U)$, computes $PWU' = h(ID_U' \| PW_U' \| a^*)$ and verifies $P_2 \stackrel{?}{=} h(PWU' \| P_1)$. Further, SC computes $B_1^* = L_1 \oplus PWU'$, $C_1 = h(ID_U \| c)$, $C_2 = h(ID_U \| a \| C_1)$, $K_{U1} = h(ID_U \| a \| P_3)$, $N = h(ID_U \| C_1 \| C_2 \| B_1^* \| T_{LA1})$, generates random number $u \in Z_q^\star$, encrypts $E_1 = E_{K_{U1}}(P_3, ID_U, C_1, C_2, u.g, N, T_{LA1})$ and sends message $M_1 = \{E_1, P_3\}$ to S via public channel.

Step LA3. On collecting M_1, S computes $K_{S1} = h(ID_U \| P_3 \oplus b \| P_3)$, decrypts $(P_3^*, ID_U^*, C_1^*, C_2^*, u.g, N^*, T_{LA1}) = D_{K_{S1}}(E_1)$ and verifies whether $P_3^* \stackrel{?}{=} P_3$ hold or not. If it does not hold, S searches $User\_List$ to find the $ID_U^* \stackrel{?}{=} ID_U$. If, it is not so, then, it does not allow new entry request and sets $Honey\_List$ to be $Honey\_List + 1$. Once the value of $Honey\_List$ eclipse the prearranged threshold, S feels that SC has been cracked. Thus, suspends SC till U registers. Otherwise, S authenticates to U. Further, S verifies the condition $T_{LA2} - T_{LA1} \leq \triangle T$, computes $B_1 = h(ID_U \| x \| c \| b)$, $N^* = h(ID_U^* \| C_1^* \| C_2^* \| B_1 \| T_{LA1})$ and again verifies whether $N^* \stackrel{?}{=} N$ hold or not. If does not hold, S sets the $Honey\_List$ to $Honey\_List + 1$ and once the amount of $Honey\_List$ outstrips the prearranged verge, suspends the SC till U re-register; Otherwise, computes $B_2 = h(ID_U^* \| x \| (P_3 \oplus b) \| b \| T_{LA1})$, selects $s \in Z_q^\star$, computes $K_{S2} = h(C_1^* \| C_2^* \| B_1 \| N^* \| T_{LA3})$, session key $SK_S = h(ID_U^* \| ID_S \| K_{S2} \| N^*.g \| u.s.g \| B_2 \| T_{LA3})$, $V = h(u.g \| u.s.g \| s.g \| T_{LA3})$, $B_2' = B_2 \oplus h(B_1 \| ID_U^* \| ID_S)$, $ID_{S1} = ID_S \oplus h(b \| N^* \| B_1 \| B_2)$, encrypts $E_2 = E_{K_{S2}}(ID_{S1}, s.g, V, B_2')$ and sends message $M_2 = \{E_2, T_{LA3}\}$ to U via public channel.

Step LA4. Upon getting $M_2 = \{E_2, T_{LA3}\}$, U verifies $T_{LA4} - T_{LA3} \leq \triangle T$. Then, computes $K_{U2} = h(C_1 \| C_2 \| B_1 \| N \| T_{LA3})$, decrypts $(ID_{S1}, s.g, V, B_2') = D_{K_{U2}}(E_2)$, computes $B_2^* = B_2' \oplus h(B_1^* \| ID_U \| ID_S^*)$ and verifies $V \stackrel{?}{=} h(u.g \| u.s.g \| s.g \| T_{LA3})$,

Table 4
Login and authentication phase.

User/ Smart card $\{G, g, h(\cdot), L_1, P_1, P_2, P_3\}$	Server S
Inputs ID'_U and PW'_U	
$a^* = P_1 \oplus h(ID_U \| PW_U)$	
$PWU' = h(ID'_U \| PW'_U \| a^*)$	
$P_2 \stackrel{?}{=} h(PWU' \| P_1)$	
$B_1^* = L_1 \oplus PWU'$	
$C_1 = h(ID_U \| c)$	
$C_2 = h(ID_U \| a \| C_1)$	
$K_{U1} = h(ID_U \| a \| P_3)$	
$N = h(ID_U \| C_1 \| C_2 \| B_1^* \| T_{LA1})$	$K_{S1} = h(ID_U \| P_3 \oplus b \| P_3)$
$u \in Z_q^*$	$(P_3^*, ID_U^*, C_1^*, C_2^*, u.g, N^*, T_{LA1}) = D_{K_{S1}}(E_1)$
$E_1 = E_{K_{U1}}(P_3, ID_U, C_1, C_2, u.g, N, T_{LA1})$	$P_3^* \stackrel{?}{=} P_3$
$M_1 = \{E_1, P_3\}$	$T_{LA2} - T_{LA1} \leq \triangle T$
. . . . . →	$B_1 = h(ID_U \| x \| c \| b)$
(via public channel)	$N^* = h(ID_U^* \| C_1^* \| C_2^* \| B_1 \| T_{LA1})$
	$N^* \stackrel{?}{=} N$
	$B_2 = h(ID_U^* \| x \| (P_3 \oplus b) \| b \| T_{LA1})$
	$s \in Z_q^*$
	$K_{S2} = h(C_1^* \| C_2^* \| B_1 \| N^* \| T_{LA3})$
	$SK_S = h(ID_U^* \| ID_S \| K_{S2} \| N^*.g \| u.s.g \| B_2 \| T_{LA3})$
	$V = h(u.g \| u.s.g \| s.g \| T_{LA3})$
	$B'_2 = B_2 \oplus h(B_1 \| ID_U^* \| ID_S)$
	$ID_{S1} = ID_S \oplus h(b \| N^* \| B_1 \| B_2)$
	$E_2 = E_{K_{S2}}(ID_{S1}, s.g, V, B'_2)$
$T_{LA4} - T_{LA3} \leq \triangle T$	$M_2 = \{E_2, T_{LA3}\}$
$K_{U2} = h(C_1 \| C_2 \| B_1 \| N \| T_{LA3})$	← . . . . .
$(ID_{S1}, s.g, V, B'_2) = D_{K_{U2}}(E_2)$	(via public channel)
$B_2^* = B'_2 \oplus h(B_1^* \| ID_U \| ID_S^*)$	
$V \stackrel{?}{=} h(u.g \| u.s.g \| s.g \| T_{LA3})$	
$ID_S^* = ID_{S1} \oplus h((P_3 \oplus a) \| N \| B_1^* \| B_2^*)$	
$SK_U = h(ID_U \| ID_S^* \| K_{U2} \| N.g \| u.s.g \| B_2^* \| T_{LA3})$	

then U authenticates to S. Moreover, U computes $ID_S^* = ID_{S1} \oplus h((P_3 \oplus a) \| N \| B_1^* \| B_2^*)$ and session key $SK_U = h(ID_U \| ID_S^* \| K_{U2} \| N.g \| u.s.g \| B_2^* \| T_{LA3})$.

Hence, mutual authentication process is success between U and S. Further, they agree to session key $SK = SK_U = SK_S$. The process of LAP is shown in Table 4.

6.4. Password change phase

When U needs to replace his/her password, she/he wants implements the followings steps:

Step PC1. U inserts SC in to card reader.

Step PC2. SC inputs ID'_U, PW'_U and computes $a^* = P_1 \oplus h(ID'_U \| PW'_U)$, $PWU' = h(ID'_U \| PW'_U \| a^*)$ and verifies whether $A'_1 = ?A_1$. If does not hold, SC eliminates the session. Otherwise, executes $B_1^* = L_1 \oplus PWU'$ and verifies $P_2 \stackrel{?}{=} h(PWU' \| P_1)$. Then, U selects the password PW_U^{NEW}. Further, U computes $a^{NEW} = P_1 \oplus h(ID_U \| PW_U^{NEW})$, $PWU^{NEW} = h(ID_U \| PW_U^{NEW} \| a^{NEW})$, $P_1^{NEW} = a^{NEW} \oplus h(ID_U \| PW_U^{NEW})$ and $L^{NEW} = B_1 \oplus PWU^{NEW}$.

Step PC3. U replaces PW_U by PW_U^{NEW}, P_1 by P_1^{NEW} and L^{NEW} by L.

6.5. Revocation phase

If U finds that his/her SC is breached or lost, he/she revoke the account without replacing the identifier as below:

Step RE1. U enters SC into card reader, takes ID'_U and PW'_U.

Step RE2. SC generates random number $a^* = P_1 \oplus h(ID_U \| PW_U)$, computes $PWU' = h(ID'_U \| PW'_U \| a^*)$ and verifies $P_2 \stackrel{?}{=} h(PWU' \| P_1)$. Further, SC computes $B_1^* = L_1 \oplus PWU'$, $C_1 = h(ID_U \| c)$, $C_2 = h(ID_U \| a \| C_1)$, $K_{U1} = h(ID_U \| a \| P_3)$, $N = h(ID_U \| C_1 \| C_2 \| B_1^* \| T_{LA1})$, generates random number $u \in Z_q^*$, encrypts $E_1 = E_{K_{U1}}(P_3, ID_U, C_1, C_2, u.g, N, T_{LA1})$ and sends message $M_{Rev1} = \{E_1, P_3, Revoke_{Request}, T_{Rev1}\}$ to S via public channel.

Step RE3. Upon getting $M_{Rev1} = \{E_1, P_3, Revoke_{Request}, T_{Rev1}\}$, S verifies $T_{Rev2} - T_{Rev1} \leq \triangle T$. Then, S computes $K_{S1} = h(ID_U \| P_3 \oplus b \| P_3)$, decrypts $(P_3^*, ID_U^*, C_1^*, C_2^*, u.g, N^*, T_{LA1}) = D_{K_{S1}}(E_1)$ and verifies whether $P_3^* \stackrel{?}{=} P_3$ hold or not. If it does not hold, S searches $User\_List$ to find the $ID_U^* \stackrel{?}{=} ID_U$. If, it is not so, then it refuses the entry request and sets $Honey\_List$ to be $Honey\_List + 1$. Once the value of $Honey\_List$ eclipse the prearranged threshold, S feels that SC has been cracked. Thus, suspends SC till U registers. Otherwise, S authenticates to U. Further, S verifies the condition $T_{LA2} - T_{LA1} \leq \triangle T$ and computes $B_1 = h(ID_U \| x \| c \| b)$, $N^* = h(ID_U^* \| C_1^* \| C_2^* \| B_1 \| T_{LA1})$ and again verifies whether $N^* \stackrel{?}{=} N$ hold or not. If does not hold, S sets the $Honey\_List$ to $Honey\_List + 1$ and once the amount of $Honey\_List$ outstrips the prearranged verge, suspends the SC till U re-register; Otherwise, S sets b to be $NULL$, thus next time, U cannot login strongly.

6.6. Re-registration phase

If the registered account of U does not work in proper manner then, U may need to re-register as the following steps:

Step RR1. U sends message $M_{RR} = \{PWU, ID_U, T_{RR1}\}$ to S.

Step RR2. On receiving M_{RR}, S checks weather whether $T_{RR2} - T_{RR1} \leq \triangle T$ hold or not. It, it does not hold, S eliminates the request and verifies the ID_U is in the $User\_List$ and whether the account of U is revoked or his/her SC is suspended. If so, S execute the re-registration as discussed in phase 2.

7. Security analysis

In this session, we have discussed security investigation of ESEAP in the following way:

7.1. Formal security analysis

Theorem: *The presented protocol employees an additive cyclic group G with generator g of order q. Under the assumption that the hash value outputs a digest of length l bits and performs as a true random oracle. Therefore, we have*

$$ADV^{ESEAP}_{\mathcal{A},\ succ} \leq \frac{q_h^2}{2^l} + \frac{q_s}{2^{l-1}} + \frac{(q_s+q_e)^2}{2^{l+1}} + 2q_h\left(ADV^{ECCDHP}_{\mathcal{A},\ succ}(q)\right) + \frac{2q_s}{X} + \frac{2q_s}{Y}. \quad (1)$$

where $ADV^{Proposed}_{\mathcal{A},\ succ}$ be the probability of success for a probabilistic polynomial time bounded, $\mathcal{A}$ is cracking the semantic security of the presented framework and $ADV^{ECCDHP}_{\mathcal{A},\ succ}$ is denoted as the probability of success $\mathcal{A}$ of solving the ECCDHP. Here, X expresses the password dictionary and Y denotes the identity dictionary. $\mathcal{A}$ asks q_s times Send queries, q_e times Execute queries and q_h times H queries to breach the security of the presented framework.

Proof: We assume that $\mathcal{A}$ can crack the semantic safety of the ESEAP, then it is probable to establish a polynomial time bounded algorithm β that can determine the ECCDHP [65], i.e., β results *r.s.g* within polynomial time bound from a random precedent (*g, u.g, s.g*), where $u, s \in Z_q^\star$. In this demonstration, we describe a sequence of games $Game_i (0 \leq i \leq 5)$ [58,59] and $\mathcal{A}$ can execute the actual attack against the proposed protocol by executing $Game_0$ where as $\mathcal{A}$ has no protection in simulation of the game $Game_5$. We also define an event $\xi_i (0 \leq i \leq 5)$ that $\mathcal{A}$ wins the $Game_i$ in breaching the semantic security of the ESEAP. Consider that the event *Z*, which is separate of ξ_i, may appear during $\mathcal{A}$'s computation such that β detects *Z*. Both $Game_i$ and $Game_{i+1}$ are indistinguishable unless *Z* occurs. Thus, we have

$$| Prob[\xi_{i+1}] - Prob[\xi_i] | \leq Prob[Z] \quad (2)$$

$Game_0$: The execution of $Game_0$ is comparable to the exact attack in the random oracle security model. Here, all the occurrence of *U* and *S* are modeled as computing in the random oracle. When $Game_0$ is computed, $\mathcal{A}$ can exactly guesses the bit τ associated in the Test query and therefore we have

$$ADV^{ESEAP}_{\mathcal{A},succ} = | 2Prob[\xi_0] - 1 | \quad (3)$$

$Game_1$: Here, this game is similar to the earlier game, except that $\mathcal{A}$ calculates the hash oracle *H* by managing a list L_H^P, which combines the tuples of the form (*Hin, Hout*). If $\mathcal{A}$ made a hash query for the input Hin_{NEW}, β returns $Hout_{NEW}$, produced a tuple (Hin_{NEW}, $Hout_{NEW}$) must be in L_H^P. Otherwise, β randomly prefers a number $Hout_{NEW} \in F_q^\star$, returns to $\mathcal{A}$ and consolidates the new tuple (Hin_{NEW}, $Hout_{NEW}$) to L_H^P. This execution of the *Execute, Reveal, Send, Corrupt* and *Test queries* are equivalent as accomplished to execute the exact attack. Thus, we have

$$Prob[\xi_1] = Prob[\xi_0] \quad (4)$$

$Game_2$: The approaching of this is similar as preceding game except that it will be exit if a collision appears in the simulation of the contents $M_1 = \{E_1, P_3\}$ and $M_2 = \{E_2, T_{LA3}\}$ based on the birthday attack. Probability of collisions of the simulation of hash oracle is at most $\frac{q_h^2}{2q}$. Alike, the probability of collisions in the contents simulation is at most $\frac{(q_s+q_e)^2}{2^{l+1}}$. Thus, we have

$$| Prob[\xi_2] - Prob[\xi_1] | \leq \frac{q_h^2}{2q} + \frac{(q_s+q_e)^2}{2^{l+1}} \quad (5)$$

$Game_3$: Here, $\mathcal{A}$ accurately guessed the authentication attributes C_1, C_2, *N*, K_{U1}, *V* and B_2^* without querying the oracle *H*. Here, this game and the last game are identical unless the user occupant or server occupant refuses a legitimate authenticated value. Thus, we have

$$| Prob[\xi_3] - Prob[\xi_2] | \leq \frac{q_s}{2^l} \quad (6)$$

$Game_4$: Here, the session key is guessed without querying the hash comparable oracle *H*, i.e., $SK_U = SK_S = SK$ is fully separate from hash oracle and *u.s.g*. Therefore, this game simulates the execution using the random self-reducibility of the *ECCDHP*. To do so, $\mathcal{A}$ queries with the random tuple (*g, u.g, s.g*) to compute $ECCDHP(u.g, s.g) = usg$, where $u, s \in Z_q^\star$.Therefore, we have

$$| Prob[\xi_4] - Prob[\xi_3] | \leq q_h ADV^{ECCDHP}_{\mathcal{A},\ succ}(q) \quad (7)$$

$Game_5$: This game is same as last game without extra *Test query* this game is exit if $\mathcal{A}$ asks a *Hquery* with $\{ID_U, ID_S^*, K_{U2}, N.g, u.s.g, B_2^*, T_{LA3}\}$. $\mathcal{A}$ can get session key $SK_U = SK_S = SK$ by executing the *Hquery* with probability at most $\frac{q_h^2}{2l}$. Thus, we have

$$| Prob[\xi_5] - Prob[\xi_4] | \leq \frac{q_h^2}{2q} \quad (8)$$

If *E* does not take any *Hquery* with the perfect input, it will not contain any improvement in differentiate the exact session key from a random one and thus $Prob[\xi_5] = \frac{1}{2}$.

Moreover, if the *Corrupt*(*U*, 2) *query* has been made, it mentions that the password *corrupt query* (*Corrupt*(*U*, 1)) has not been made. The probability of $\mathcal{A}$ launching the off-line password guessing attacks are $\frac{q_s}{X}$ and the probability of $\mathcal{A}$ launching the off-line identity guessing attacks are $\frac{q_s}{Y}$. Thus, from the equations (3) − (8), we obtained

$$ADV^{ESEAP}_{\mathcal{A},\ succ} \leq \frac{q_h^2}{2^l} + \frac{q_s}{2^{l-1}} + \frac{(q_s+q_e)^2}{2^{l+1}} + 2q_h\left(ADV^{ECCDHP}_{\mathcal{A},\ succ}(q)\right) + \frac{2q_s}{X} + \frac{2q_s}{Y}. \quad (9)$$

7.2. Informal security analysis

Here, we have discussed security attributes and features which are verified by ESEAP as following:

1. **Off-line password guessing attack:** Suppose that $\mathcal{A}$ extracted the data in *SC* and eavesdropped the *U* and *S*, he/she cannot archive the attack as explained below:
   a. $\mathcal{A}$ guesses the $PW_\mathcal{A}$ and $ID_\mathcal{A}$, generates a random value $a_\mathcal{A} \in Z_q^\star$ and executes $PWE = h(ID_\mathcal{A} \| PW_\mathcal{A} \| a_\mathcal{A})$.
   b. Verify the exactness of *PWU* and ID_U by the verifying the condition $P_2 \stackrel{?}{=} h(PW\mathcal{A} \| P_1)$. For this, $\mathcal{A}$ tests $PW_\mathcal{A} \stackrel{?}{=} PW_U, PW\mathcal{A} \stackrel{?}{=} PWU$ and $ID_U \stackrel{?}{=} ID_\mathcal{A}$ while there are uncertain parameters $PW\mathcal{A}, ID_\mathcal{A}, a_\mathcal{A}$ and $PW_\mathcal{A}$. The $PW\mathcal{A}$ and $ID_\mathcal{A}$ are in a finite values but value $a_\mathcal{A}$ is too hard to be guessed.

   Thus, ESEAP has resistance to this attack.
2. **Smart card loss attack**
   Suppose that $\mathcal{A}$ stolen *U*'s *SC* and obtained the values. According to 'off-line password guessing attack', $\mathcal{A}$ cannot try an off-line guessing password. On the other hand, If possible $\mathcal{A}$ guesses $PW\mathcal{A} = PWU$ and $ID_\mathcal{A} = ID_U$, computes $B_{\mathcal{A}1}^* = L_1 \oplus PW\mathcal{A}$, $C_{\mathcal{A}1} = h(ID_U \| c)$, $C_{\mathcal{A}2} = h(ID_U \| a \| C_{\mathcal{A}1})$,

$K_{U\mathcal{A}} = h(ID_U \| a \| P_3)$, $N_{\mathcal{A}} = h(ID_U \| C_{\mathcal{A}1} \| C_{\mathcal{A}2} \| B^*_{\mathcal{A}1} \| T_{LA1})$, $u_{\mathcal{A}} \in Z^*_q$, $E_{\mathcal{A}1} = E_{K_{U\mathcal{A}}}(P_{\mathcal{A}3}, ID_{\mathcal{A}}, C_{\mathcal{A}1}, C_{\mathcal{A}2}, u_{\mathcal{A}}.g, N_{\mathcal{A}}, T_{LA1})$ and sends $M_{\mathcal{A}1} = \{E_{\mathcal{A}1}, P_3\}$ to S over public channel. On collecting message, S checks the condition $T_{LA2} - T_{LA\mathcal{A}} \leq \triangle T$. If possible hold, S Computes $K_{S1} = h(ID_U \| P_3 \oplus b \| P_3)$ and decrypts $(P_{\mathcal{A}3}, ID_{\mathcal{A}}, C_{\mathcal{A}1}, C_{\mathcal{A}2}, u_E.g, N_{\mathcal{A}}, T_{LA1}) = D_{K_{S1}}(E_{\mathcal{A}1})$. Here, $P_{\mathcal{A}3} \neq P_3$ and $N^* \neq N_{\mathcal{A}}$. Hence, the ESEAP secure against smart card loss attack.

3. **Support anonymity**
   In ESEAP, S used the anonymous identity ID^*_U of U and S used the anonymous identity ID^*_S of S in LAP. Thus, identity is untraceable. Therefore, ESEAP preserves the identities.
4. **Mutual authentication**
   In LAP, S computes $V = h(u.g \| u.s.g \| s.g \| T_{LA3})$ and U computes and verifies $V \stackrel{?}{=} h(u.g \| u.s.g \| s.g \| T_{LA3})$. Therefore, U and S mutual authenticated to each other. Hence, ESEAP achieved mutual authentication property.
5. **Replay attack**
   The ESEAP takes advantage of time-stamp and random values to prohibit this attack. In login and authentication phase, time-stamps condition $T_i - T_j \leq \triangle T$, where $\triangle T$ is the maximum time delay. In every steps U/S used the different random numbers. If E replay the eavesdropped message from the public channel, he/she verifies $N^* \stackrel{?}{=} N$ and $V \stackrel{?}{=} h(u.g \| u.s.g \| s.g \| T_{LA3})$. Further, E finds session key $SK = SK_U = SK_S$, which is not possible. Thus, the proposed framework secure against reply attack.
6. **User impersonation attack**
   To impersonate an appropriate U, $\mathcal{A}$ commonly has two ways: getting password PWU and identity ID_U, and constructing $M_1 = \{E_1, P_3\}$. According as off-line password guessing attack and smart card loss attack. It is not possible for $\mathcal{A}$. Hence, ESEAP secure against this attack.
7. **Server spoofing attack**
   As x is a security key for server, $\mathcal{A}$ can not recover $B_1 = h(ID_U \| x \| c \| b)$ and $N^* = h(ID^*_U \| C^*_1 \| C^*_2 \| B_1 \| T_{LA1})$. Here, $N^* \neq N$. Thus, $\mathcal{A}$ can not impersonate to S.
8. **De-synchronization attack**
   There are no parameters wants to be update in U and S side; on the other side, whenever U needs to replace her/his password, he/she to validated to each other. Lastly, the proposed scheme does not require the U or S to synchronization property. Thus, ESEAP secure against this attack.
9. **Provision of key agreement**
   In ESEAP, U and S authenticate to each other with $B^*_1 = B_1, B^*_2 = B_2, K_{S2} = K_{U2}, ID^*_U = ID_1, ID^*_S = ID_S$ and $N^* = N$. So, they have agreed to session key $SK = SK_S = SK_U$.
10. **Insider attack:** In registration phase, U submits $PWU = h(ID_U \| PW_U \| a)$, where a is the random value. Thus, administrator can not get PWU. Hence, ESEAP can withstand insider attack.
11. **Message authentication**
    In LAP, S receives message $M_1 = \{E_1, P_3\}$ and verified $P^*_3 = P_3$, $T_{LA2} - T_{LA1} \leq \triangle T$ and $N^* = N$, then sends message, $M_2 = \{E_2, T_{LA3}\}$ to U. On receiving M_2, U verified $T_{LA4} - T_{LA3} \leq \triangle T$ and $V = h(u.g \| u.s.g \| s.g \| T_{LA3})$. If any of the information fail, message will not be accepted. Thus, message authentication verified between U and S.

8. Performance analysis

Here, we explained the security and functionality attributes comparison, communication and computation cost comparison of the ESEAP with related frameworks like Kumari et al.'s [44], Kumari et al.'s [45], Jiang et al.'s [46], Islam et al.'s [47], Marimuthu et al.'s [48], Maitra et al.'s [49], Xie et al.'s [50], Wang et al.'s [43] Srinivas et al.'s [51], Wang et al. [38], Buyn et al.'s [32], Amin et al.'s [39], Wang et al.'s [16], Odelu et al.'s [31], Wu et al.'s [40], Ali et al.'s [41], Luo et al.'s [33] and Roy et al.'s [42]. The security analysis is very much essential to analyze the behavior of ESEAP and compare with the earlier proposed schemes in the same environment. The comparison of the security features is required to show the resistance and preservation of the attacks which proves the scheme is secure or insecure. The more details are shown in Fig. 1. Further, the comparison of the computations costs for the above mentioned schemes and ESEAP is required to show the time complexity in perform LAP. Finally, the comparison of the communication cost in bits is required to transmit the bits among the communicated parties. Therefore, the evaluation manages an insight in effectiveness of ESEAP with other schemes.

8.1. Security and functionality features comparison

Table 5 shows that the security and functionality attributes of ESEAP with other frameworks like Kumari et al.'s [44], Kumari et al.'s [45], Jiang et al.'s [46], Islam et al.'s [47], Marimuthu et al.'s [48], Maitra et al.'s [49], Xie et al.'s [50], Wang et al.'s [43] Srinivas et al.'s [51], Wang et al. [38], Buyn et al.'s [32], Amin et al.'s [39], Wang et al.'s [16], Odelu et al.'s [31], Wu et al.'s [40], Ali et al.'s [41], Luo et al.'s [33] and Roy et al.'s [42]. In summary, ESEAP manages many functionality attributes and it is also safe against possible known security attacks for secure communication.

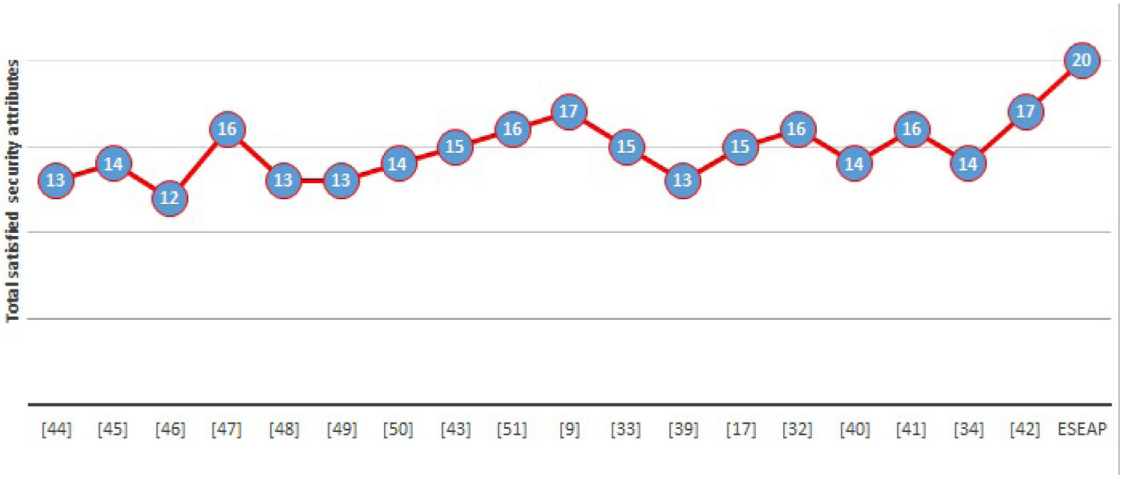

Fig. 1. Security and functionality features comparison.

Table 5
Security and functionality features comparison.

Security Attack	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}	P_{19}	P_{20}	**T**
Kumari et al. [44]	Y	N	Y	Y	Y	Y	N	N	Y	N	Y	Y	N	Y	N	Y	Y	Y	N	Y	13
Kumari et al. [45]	Y	N	N	Y	Y	Y	Y	Y	Y	N	Y	N	Y	Y	N	Y	Y	Y	N	Y	14
Jiang et al. [46]	Y	Y	N	Y	Y	Y	Y	N	Y	N	Y	Y	Y	Y	N	N	N	Y	N	N	12
Islam et al. [47]	Y	N	N	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	16
Marimuthu et al. [48]	Y	N	Y	Y	Y	N	Y	N	N	×	N	Y	Y	Y	N	Y	Y	Y	Y	Y	13
Maitra et al. [49]	Y	N	Y	Y	Y	Y	N	Y	Y	N	N	Y	Y	Y	N	N	Y	Y	N	Y	13
Xie et al. [50]	Y	Y	N	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	N	Y	N	N	14
Wang et al. [43]	N	N	Y	Y	Y	N	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	N	Y	15
Srinivas et al. [51]	Y	Y	Y	Y	Y	N	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	N	16
Wang et al. [38]	Y	Y	N	Y	Y	Y	N	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	17
Buyn et al. [32]	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N	N	Y	N	Y	15
Amin et al. [39]	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	Y	N	N	Y	13
Wang et al. [16]	Y	Y	N	Y	Y	Y	N	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	15
Odelu et al. [31]	Y	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	16
Wu et al. [40]	Y	N	N	Y	N	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	N	Y	N	14
Ali et al. [41]	Y	N	N	Y	Y	Y	Y	N	Y	N	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	16
Luo et al. [33]	Y	N	N	Y	N	Y	Y	Y	Y	N	Y	Y	Y	N	N	Y	Y	Y	Y	N	14
Roy et al. [42]	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	N	Y	Y	Y	17
ESEAP	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	20

Note $\Longrightarrow$ *Y*: Attribute is satisfied by the scheme, *N*: Attribute is not satisfied by the scheme, × : Attribute is not related to scheme, **T**:Total satisfied attribute, P_1:Off-line password guessing attack, P_2:Smart card loss attack, P_3:User anonymity, P_4:Mutual authentication, P_5:Replay attack, P_6:User impersonation attack, P_7:Server spooling attack, P_8:No clock-synchronization attack, P_9:Provision of key agreement, P_{10}:Forward secrecy, P_{11}:Insider attack, P_{12}:Message authentication, P_{13}:Parallel session attack, P_{14}: No password verifier-table, P_{15}: Sound repairability, P_{16}: No password exposure, P_{17}: Timely typo detection, P_{18}: Resistance to know attacks, P_{19}: Password friendly and P_{20}:User and server unlinkability.

Table 6
Computation cost for various operations.

Notations	Descriptions	Time (Second)
T_{Hash}	The time for calculating one-way hash function	≈ 0.00032
T_{Exp}	The time of modular exponentiation operation	≈ 0.0192
T_{Mud}	The running time of modular multiplication/division operation	≈ 0.00264
T_{Eccm}	The running time of elliptic curve point multiplication	≈ 0.01771
$T_{E/D}$	The running time of symmetric encryption/decryption	≈ 0.0056
T_{Sign}	The time for computing execute/verify a signature	≈ 0.2182
T_{CRT}	Time to find solution using Chinese remainder theorem	≈ 0.00704

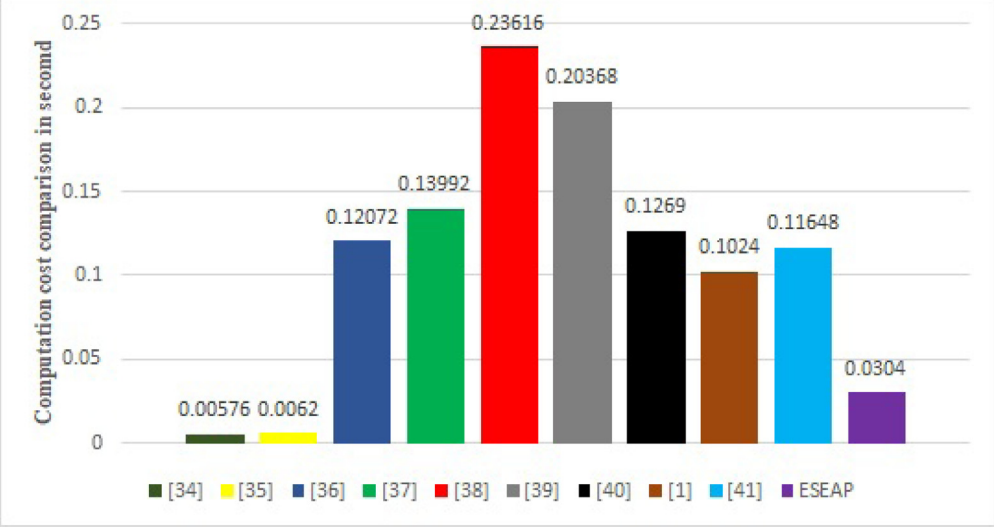

Fig. 2. Computation cost comparison.

8.2. Computation cost comparison

In this section, we discussed the computation cost comparison of ESEAP with exist relevant frameworks such as Kumari et al.'s [44], Kumari et al.'s [45], Jiang et al.'s [46], Islam et al.'s [47], Marimuthu et al.'s [48], Maitra et al.'s [49], Xie et al.'s [50], Wang et al.'s [43] Srinivas et al.'s [51], Wang et al. [38], Buyn et al.'s [32], Amin et al.'s [39], Wang et al.'s [16], Odelu et al.'s [31], Wu et al.'s [40], Ali et al.'s [41], Luo et al.'s [33] and Roy et al.'s [42]. The analysis manages an insight performance of ESEAP with other frameworks. We adopted the computation cost as experimental results of [51,66–68]. The computation cost of the cryptographic components and its cost shown Table 6.

From Table 7, the computation cost of ESEAP is $25T_{Hash} + 4T_{E/D} \approx 0.0304$ Second which is greater than Kumari et al.'s [44], Kumari et al.'s [45], Jiang et al.'s [46], Islam et al.'s [47], Marimuthu

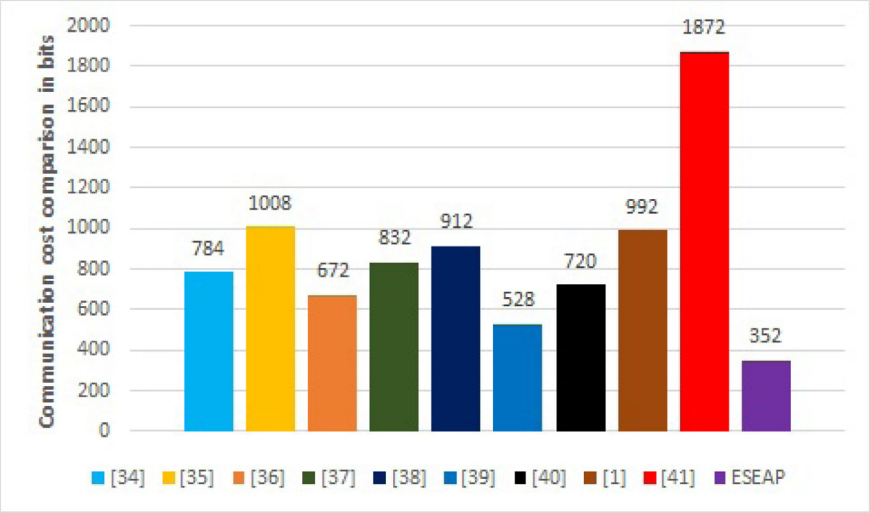

Fig. 3. Communication cost comparison.

Table 7
Comparison of computation cost with related protocols.

Protocols	Total computation cost	Time(Second)
Kumari et al. [44]	$18T_{Hash}$	≈ 0.00576
Kumari et al. [45]	$21T_{Hash}$	≈ 0.0062
Jiang et al. [46]	$9T_{Hash} + 6T_{Exp} + 1T_{Mud}$	≈ 0.12072
Islam et al. [47]	$9T_{Hash} + 7T_{Exp} + 1T_{Mud}$	≈ 0.13992
Marimuthu et al. [48]	$18T_{Hash} + 12T_{Exp}$	≈ 0.23616
Maitra et al. [49]	$20T_{Hash} + 10T_{Exp} + 2T_{Mud}$	≈ 0.20368
Xie et al. [50]	$12T_{Hash} + 6T_{Eccm} + 3T_{E/D}$	≈ 0.1269
Wang et al. [43]	$20T_{Hash} + 5T_{Exp}$	≈ 0.1024
Srinivas et al. [51]	$42T_{Hash} + 5T_{Exp} + 1T_{CRT}$	≈ 0.11648
Wang et al. [38]	$42T_{Hash} + 5T_{Exp}$	≈ 0.10944
Buyn et al. [32]	$8T_{Hash} + 2T_{Exp} + 10T_{Mud} + 2T_{E/D}+$	≈ 0.07856
Amin et al. [39]	$26T_{Hash}/21T_{Hash}$	≈ 0.00832/0.00672
Wang et al. [16]	$12T_{Hash} + 2T_{E/D} + 3T_{Mud}$	≈ 0.02296
Odelu et al. [31]	$8T_{Hash} + 2T_{E/D} + 8T_{Mud}$	≈ 0.03488
Wu et al. [40]	$26T_{Hash} + 4T_{E/D} + 4T_{Exp}$	≈ 0.1824
Ali et al. [41]	$24T_{Hash} + 8T_{E/D}$	≈ 0.05248
Luo et al. [33]	$26T_{Hash}$	≈ 0.00832
Roy et al. [42]	$20T_{Hash} + 2T_{E/D} + 1T_{Mud}$	≈ 0.02024
ESEAP	$16T_{Hash} + 6T_{Mud}$	≈ 0.0304

Table 8
Comparison of communication cost with related protocols.

Protocols	No of messages	Communication cost (bits)
Kumari et al. [44]	2	784
Kumari et al. [45]	3	1008
Jiang et al. [46]	2	672
Islam et al. [47]	2	832
Marimuthu et al. [48]	3	912
Maitra et al. [49]	2	528
Xie et al. [50]	3	720
Wang et al. [43]	3	992
Srinivas et al. [51]	3	1872
Wang et al. [38]	3	832
Buyn et al. [32]	2	576
Amin et al. [39]	3	1440/1376
Wang et al. [16]	3	784
Odelu et al. [31]	3	736
Wu et al. [40]	3	1936
Ali et al. [41]	3	1584
Luo et al. [33]	3	1792
Roy et al. [42]	2	928
ESEAP	2	352

et al.'s [48], Maitra et al.'s [49], Xie et al.'s [50], Wang et al.'s [43] Srinivas et al.'s [51], Wang et al. [38], Buyn et al.'s [32], Amin et al.'s [39], Wang et al.'s [16], Odelu et al.'s [31], Wu et al.'s [40], Ali et al.'s [41], Luo et al.'s [33] and Roy et al.'s [42]. Also, the proposed protocol is more secure as with others protocols. The details of computation cost of ESEAP and other relevant frameworks is shown in Fig. 2.

8.3. Communication cost comparison

In ESEAP, we take communication cost from Mohit et al.'s [69] scheme. The details of operations and communication cost like the communication cost of identity, time-stamp, generated number of 48 bits; asymmetric encryption/ decryption operation, symmetric encryption/ decryption operation, and modular multiplication/ inversion operation of 128 bits; bilinear pairing and cryptographic hash function of 160 bits and executing/verifying a signature of 512 bits. We summarize the communication cost in Table 8, the communication cost of ESEAP is 352 bits, the communication cost the related frameworks Kumari et al.'s [44], Kumari et al.'s [45], Jiang et al.'s [46], Islam et al.'s [47], Marimuthu et al.'s [48], Maitra et al.'s [49], Xie et al.'s [50], Wang et al.'s [43] Srinivas et al.'s [51], Wang et al. [38], Buyn et al.'s [32], Amin et al.'s [39], Wang et al.'s [16], Odelu et al.'s [31], Wu et al.'s [40], Ali et al.'s [41], Luo et al.'s [33] and Roy et al.'s [42]. The details of communication cost is shown in Table 8. The ESEAP is efficient in terms of communication cost and ESEAP can resist all the above security attributes. The communication cost of ESEAP and other relevant framework is displayed in Fig. 3.

9. Future directions and conclusion

The paper demonstrates various security drawbacks of Wang et al. scheme such as off-line password guessing attack and impersonation attack. The paper has advised a solution by designing an ECC based secure and efficient mutual authentication protocol using smart card. Further, we proved that the suggested framework provides better security attributes and functionality features than related schemes. Furthermore, we provided a formal security proof of the ESEAP, which based on random oracle model. The presented protocol is also more efficient in terms of computation and communication cost with relevant protocols in the same environment. Hence, the proposed protocol is a real-life application in communication system.

Declaration of Competing Interest

None.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.jisa.2019.102443.

References

[1] Xia Z, Wang X, Zhang L, Qin Z, Sun X, Ren K. A privacy-preserving and copy--deterrence content-based image retrieval scheme in cloud computing. IEEE Trans Inf Forens Secur 2016;11(11):2594–608.
[2] Fu Z, Ren K, Shu J, Sun X, Huang F. Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Trans Parallel Distrib Syst 2016;27(9):2546–59.
[3] Lamport L. Password authentication with insecure communication. Commun ACM 1981;24(11):770–2.
[4] Shimizu A. A dynamic password authentication method using a one-way function. Syst Comput Japan 1991;22(7):32–40.
[5] Shieh S-P, Yang W-H, Sun H-M. An authentication protocol without trusted third party. IEEE Commun Lett 1997;1(3):87–9.
[6] Chang C-C, Wu T-C. Remote password authentication with smart cards. IEE Proc E (Comput Digital Tech) 1991;138(3):165–8.
[7] He D, Zeadally S, Xu B, Huang X. An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. IEEE Trans Inf Forens Secur 2015;10(12):2681–91.
[8] Yu J, Ren K, Wang C. Enabling cloud storage auditing with verifiable outsourcing of key updates. IEEE Trans Inf Forens Secur 2016;11(6):1362–75.
[9] Fan C-I, Chan Y-C, Zhang Z-K. Robust remote authentication scheme with smart cards. Computers & Security 2005;24(8):619–28.
[10] Das ML, Saxena A, Gulati VP. A dynamic id-based remote user authentication scheme. IEEE Trans Consum Electron 2004;50(2):629–31.
[11] Juang W-S, Chen S-T, Liaw H-T. Robust and efficient password-authenticated key agreement using smart cards. IEEE Trans Ind Electron 2008;55(6):2551–6.
[12] Mishra D, Kumar V, Mukhopadhyay S. A pairing-free identity based authentication framework for cloud computing. In: International conference on network and system security. Springer; 2013. p. 721–7.
[13] Sun D-Z, Huai J-P, Sun J-Z, Li J-X, Zhang J-W, Feng Z-Y. Improvements of Juang's password-authenticated key agreement scheme using smart cards. IEEE Trans Ind Electron 2009;56(6):2284–91.
[14] Li X, Qiu W, Zheng D, Chen K, Li J. Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. IEEE Trans Ind Electron 2010;57(2):793–800.
[15] Chaudhry SA, Farash MS, Naqvi H, Kumari S, Khan MK. An enhanced privacy preserving remote user authentication scheme with provable security. Secur Commun Netw 2015;8(18):3782–95.
[16] Wang D, Wang N, Wang P, Qing S. Preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity. Inf Sci 2015;321:162–78.
[17] Kumar V, Jangirala S, Ahmad M. An efficient mutual authentication framework for healthcare system in cloud computing. J Med Syst 2018;42(8):142.
[18] Huang X, Chen X, Li J, Xiang Y, Xu L. Further observations on smart-card-based password-authenticated key agreement in distributed systems. IEEE Trans Parallel Distrib Syst 2013;25(7):1767–75.
[19] Juang W-S, Chen S-T, Liaw H-T. Robust and efficient password-authenticated key agreement using smart cards. IEEE Trans Ind Electron 2008;55(6):2551–6.
[20] Sun D-Z, Huai J-P, Sun J-Z, Li J-X, Zhang J-W, Feng Z-Y. Improvements of Juang's password-authenticated key agreement scheme using smart cards. IEEE Trans Ind Electron 2009;56(6):2284–91.
[21] Wang D, Wang P. On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions. Comput Netw 2014;73:41–57.
[22] Fan R, He D-j, Pan X-z, et al. An efficient and dos-resistant user authentication scheme for two-tiered wireless sensor networks. J Zhejiang Uni Sci C 2011;12(7):550–60.
[23] Xue K, Ma C, Hong P, Ding R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. J Netw Comput Appl 2013;36(1):316–23.
[24] Chuang M-C, Lee J-F, Chen M-C. Spam: a secure password authentication mechanism for seamless handover in proxy mobile ipv6 networks. IEEE Syst J 2012;7(1):102–13.
[25] Wang D, He D, Wang P, Chu C-H. Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. IEEE Trans Depend Secure Comput 2014;12(4):428–42.
[26] Li C-T. A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card. IET Inf Secur 2013;7(1):3–10.
[27] Tsai J-L, Lo N-W, Wu T-C. Novel anonymous authentication scheme using smart cards. IEEE Trans Ind Inf 2012;9(4):2004–13.
[28] Wang D, Gu Q, Cheng H, Wang P. The request for better measurement: a comparative evaluation of two-factor authentication schemes. In: Proceedings of the 11th ACM on Asia conference on computer and communications security. ACM; 2016. p. 475–86.
[29] Li X, Niu J, Khan MK, Liao J. An enhanced smart card based remote user password authentication scheme. J Netw Comput Appl 2013;36(5):1365–71.
[30] Kumari S, Khan MK. Cryptanalysis and improvement of a robust smart–card-based remote user password authentication schemeâ. Int J Commun Syst 2014;27(12):3939–55.
[31] Odelu V, Das AK, Goswami A. An effective and robust secure remote user authenticated key agreement scheme using smart cards in wireless communication systems. Wirel Pers Commun 2015;84(4):2571–98.
[32] Bin Muhaya FT. Cryptanalysis and security enhancement of zhu's authentication scheme for telecare medicine information system. Secur Commun Netw 2015;8(2):149–58.
[33] Luo H, Wen G, Su J. Lightweight three factor scheme for real-time data access in wireless sensor networks. Wirel Netw 2018:1–16.
[34] Ma C-G, Wang D, Zhao S-D. Security flaws in two improved remote user authentication schemes using smart cards. Int J Commun Syst 2014;27(10):2215–27.
[35] Madhusudhan R, Mittal R. Dynamic id-based remote user password authentication schemes using smart cards: a review. J Netw Comput Appl 2012;35(4):1235–48.
[36] Wang D, He D, Wang P, Chu C-H. Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. IEEE Trans Depend Secure Comput 2015;12(4):428–42.
[37] Chaudhry SA, Naqvi H, Mahmood K, Ahmad HF, Khan MK. An improved remote user authentication scheme using elliptic curve cryptography. Wirel Pers Commun 2017;96(4):5355–73.
[38] Wang D, Wang P. Two birds with one stone: two-factor authentication with security beyond conventional bound. IEEE Trans Depend Secure Comput 2016;15(4):708–22.
[39] Amin R, Islam SH, Gope P, Choo K-KR, Tapas N. Anonymity preserving and lightweight multi-medical server authentication protocol for telecare medical information system. IEEE J Biomed Health Inf 2018.
[40] Wu F, Xu L, Kumari S, Li X. A new and secure authentication scheme for wireless sensor networks with formal proof. Peer-to-Peer Netwo Appl 2017;10(1):16–30.
[41] Ali R, Pal AK, Kumari S, Karuppiah M, Conti M. A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring. Future Generat Comput Syst 2018;84:200–15.
[42] Roy S, Das AK, Chatterjee S, Kumar N, Chattopadhyay S, Rodrigues JJ. Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications. IEEE Trans Ind Inf 2018;15(1):457–68.
[43] Wang C, Wang D, Xu G, Guo Y. A lightweight password-based authentication protocol using smart card. Int J Commun Syst 2017;30(16).
[44] Kumari S, Khan MK, Li X. An improved remote user authentication scheme with key agreement. Comput Electr Eng 2014;40(6):1997–2012.
[45] Kumari S, Li X, Wu F, Das AK, Odelu V, Khan MK. A user anonymous mutual authentication protocol.. KSII Trans Internet Inf Syst 2016;10(9).
[46] Jiang Q, Ma J, Li G, Li X. Improvement of robust smart-card-based password authentication scheme. Int J Commun Syst 2015;28(2):383–93.
[47] Islam S. Design and analysis of an improved smartcard-based remote user password authentication scheme. Int J Commun Syst 2016;29(11):1708–19.
[48] Karuppiah M, Saravanan R. A secure remote user mutual authentication scheme using smart cards. J Inf Secur Appl 2014;19(4–5):282–94.
[49] Maitra T, Obaidat MS, Amin R, Islam S, Chaudhry SA, Giri D. A robust elgamal-based password-authentication protocol using smart card for client-server communication. Int J Commun Syst 2017 2017;30(11).
[50] Xie Q, Wong DS, Wang G, Tan X, Chen K, Fang L. Provably secure dynamic id-based anonymous two-factor authenticated key exchange protocol with extended security model. IEEE Trans Inf Forens Secur 2017;12(6):1382–92.
[51] Srinivas J, Das AK, Kumar N, Rodrigues J. Cloud centric authentication for wearable healthcare monitoring system. IEEE Trans Depend Secure Comput 2018.
[52] Dinarvand N, Barati H. An efficient and secure rfid authentication protocol using elliptic curve cryptography. Wirel Netw 2019;25(1):415–28.
[53] Kumar N, Kaur K, Misra SC, Iqbal R. An intelligent rfid-enabled authentication scheme for healthcare applications in vehicular mobile cloud. Peer-to-Peer Netw Appl 2016;9(5):824–40.
[54] Eisenbarth T, Kasper T, Moradi A, Paar C, Salmasizadeh M, Shalmani MTM. On the power of power analysis in the real world: acomplete break of the keeloq code hopping scheme. In: Annual International Cryptology Conference. Springer; 2008. p. 203–20.
[55] Dolev D, Yao A. On the security of public key protocols. IEEE Trans Inf Theory 1983;29(2):198–208.
[56] Cao X, Zhong S. Breaking a remote user authentication scheme for multi-server architecture. IEEE Commun Lett 2006;10(8):580–1.
[57] Bellare M, Rogaway P. Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM conference on Computer and communications security. ACM; 1993. p. 62–73.
[58] Shoup V. Sequences of games: a tool for taming complexity in security proofs.. IACR Cryptol ePrint Arch 2004;2004:332.
[59] Xu J, Zhu W-T, Feng D-G. An improved smart card based password authentication scheme with provable security. Comput Stand Interf 2009;31(4):723–8.
[60] Kocher P, Jaffe J, Jun B. Differential power analysis. In: Annual International Cryptology Conference. Springer; 1999. p. 388–97.

[61] Messerges TS, Dabbish EA, Sloan RH. Examining smart-card security under the threat of power analysis attacks. IEEE Trans Comput 2002;51(5):541–52.
[62] Joye M, Olivier F. Side-channel analysis. Encyclopedia of Cryptography and Security. Springer; 2005. 571–571
[63] Fan C-I, Lin Y-H. Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics. IEEE Trans Inf Forens Secur 2009;4(4):933–45.
[64] Islam SH, Vijayakumar P, Bhuiyan MZA, Amin R, Balusamy B, et al. A provably secure three-factor session initiation protocol for multimedia big data communications. IEEE Internet Thing J 2018;5(5):3408–18.
[65] Mishra D, Das AK, Mukhopadhyay S. A secure and efficient ecc-based user anonymity-preserving session initiation authentication protocol using smart card. Peer-to-peer Netw Appl 2016;9(1):171–92.
[66] Challa S, Wazid M, Das AK, Khan MK. Authentication protocols for implantable medical devices: taxonomy, analysis and future directions. IEEE Consum Electron Mag 2018;7(1):57–65.
[67] He D, Zeadally S. Authentication protocol for an ambient assisted living system. IEEE Commun Mag 2015;53(1):71–7.
[68] Odelu V, Das AK, Goswami A. A novel linear polynomial-based dynamic key management scheme for hierarchical access control. Int J Trust Manag Comput Commun 2013;1(2):156–74.
[69] Mohit P, Amin R, Karati A, Biswas G, Khan MK. A standard mutual authentication protocol for cloud computing based health care system. J Med Syst 2017;41(4):50.