applied sciences

MDPI

Article

Elliptic Curve Signcryption-Based Mutual Authentication Protocol for Smart Cards

Anuj Kumar Singh [1], Arun Solanki [2], Anand Nayyar [3,4,*] and Basit Qureshi [5]

[1] Department of Computer Science & Engineering, Amity University Haryana, Gurugram 122413, India; aksingh@ggn.amity.edu
[2] Department of Computer Science & Engineering, Gautam Buddha University, Greater Noida 201312, India; asolanki@gbu.ac.in
[3] Graduate School, Duy Tan University, Da Nang 550000, Vietnam
[4] Faculty of Information Technology, Duy Tan University, Da Nang 550000, Vietnam
[5] Robotics and Internet of Things Lab, Department of Computer Science, Prince Sultan University, Riyadh 11586, Saudi Arabia; qureshi@psu.edu.sa
* Correspondence: anandnayyar@duytan.edu.vn

Received: 21 September 2020; Accepted: 17 November 2020; Published: 23 November 2020

Abstract: In the modern computing environment, smart cards are being used extensively, which are intended to authenticate a user with the system or server. Owing to the constrictions of computational resources, smart card-based systems require an effective design and efficient security scheme. In this paper, a smart card authentication protocol based on the concept of elliptic curve signcryption has been proposed and developed, which provides security attributes, including confidentiality of messages, non-repudiation, the integrity of messages, mutual authentication, anonymity, availability, and forward security. Moreover, the analysis of security functionalities shows that the protocol developed and explained in this paper is secure from password guessing attacks, user and server impersonation, replay attacks, de-synchronization attacks, insider attacks, known key attacks, and man-in-the-middle attacks. The results have demonstrated that the proposed smart card security protocol reduces the computational overhead on a smart card by 33.3% and the communication cost of a smart card by 34.5%, in comparison to the existing efficient protocols. It can, thus, be inferred from the results that using elliptic curve signcryption in the authentication mechanism reduces the computational cost and communication overhead by a significant amount.

Keywords: smart card; security; elliptic curve cryptography; authentication; signcryption; protocol

1. Introduction

The computing environment has changed rapidly in the past two decades on account of technological advancements and innovations. The modern computing environment implements the automation of tasks and processes largely. Automatic identification and data capture are one of the most important components of process automation, and are used in a variety of applications. The technologies behind automatic identification and data capture include bar codes, chip cards, biometrics, voice recognition, optical character recognition, smart cards, magnetic stripes, and radio frequency identification (RFID). Among these technologies, smart cards are being used widely in many critical applications due to their advantages of longer life, larger memory, high security, and lesser cost of operations [1]. The applications of smart cards include secure electronic payments, secure identification, mobile applications, secure physical access, and healthcare systems, to name a few. Usually, a CPU (central processing unit), ROM (read-only memory), a RAM (random access memory), a co-processor, and an electrically erasable programmable read-only memory (EEPROM) are the major components of

a smart card. The representative diagram of a smart card has been demonstrated in Figure 1. ROM is used to store the operating system (OS) and the run-time environment. The data and the instructions to be executed by the CPU are kept in RAM. The application and its data are stored in EEPROM [2]. All data processing and instructions are executed by the CPU, while the co-processor is used to enhance the performance of security functions such as encryption and decryption.

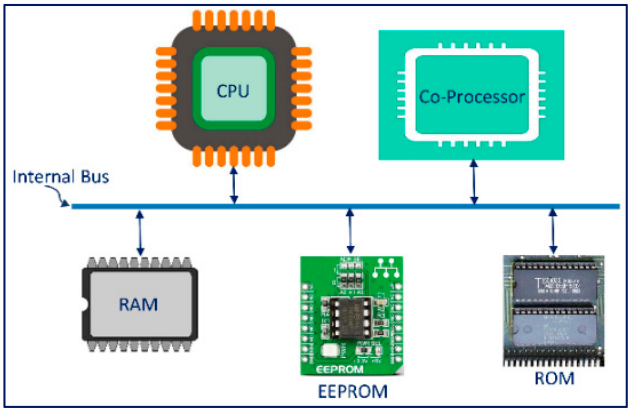

Figure 1. Architecture of a smart card. CPU: central processing unit, ROM: read-only memory, RAM: random access memory, EEPROM: electrically erasable programmable read-only memory.

1.1. Security Requirements for Smart Cards

Authentication, non-repudiation, confidentiality, and integrity are the fundamental security properties that must be satisfied by any system. However, by studying Ko and Caytiles [3] and Pippal et al. [4], it can be figured out that in addition to these four fundamental security properties, forward security, along with availability, must also be included in the implementation in order to secure the smart cards. The security attributes satisfied by smart card-based systems are briefly described below:

- Confidentiality—This property ensures that the information or data transmitted and received from the smart card are in an incomprehensible form, so that an unauthorized party cannot understand it.
- Authentication—The smart card must authenticate the other party before sending any message or information to it.
- Non-repudiation—The smart card and the other legitimate party cannot repudiate after sending or receiving the transmitted message.
- Integrity—The data sent or received by the smart card must not be rehabilitated in transit.
- Availability—The smart card must be capable to produce the required data as and when needed by the user.
- Forward Security—This attribute ensures that an adversary is not be able to obtain the past messages, even if they are in the possession of a secret session key for the long term.

1.2. Security Challenges of Smart Cards

Designing efficient security solutions for smart card-based systems has been an incessant challenge for professionals and the researchers alike, due to the following limitations:

- Low Computational Capability—The smart cards have a low computational capacity of 1–5 MHz clock rate, less than 1 MB flash memory, and a few hundreds of KB RAM [5]. Owing to these restricted computational resources, it is challenging to develop computationally efficient security solutions that satisfy the required security attributes as well as security functions.
- Less Power—Certain smart cards operate on power, which is limited. The security operations must be carefully selected in a way that avoids heavy computational operations and tasks.

- Unreliable Communication—The messages and data sent by the smart cards and received by them passes through an unreliable wireless medium, and therefore, it is more susceptible to different attacks. Strong security countermeasures must be implemented to defeat these attacks.

These constraints of smart cards pose two security challenges. The first challenge is to defeat the attacks attempted on smart cards. The second one is to implement the efficient cryptographic solutions that can provide necessary security properties and can resist the attacks on smart cards. Ko and Caytiles wrote a review of the security of smart card-based systems and categorized the attacks on smart cards as side-channel attacks, logical attacks, physical attacks, and other attacks. Pippal et al. [4] pointed out that impersonation attacks, session attacks, password-guessing attacks, and replay attacks can be launched on smart cards. Mahanta et al. [6] have explored that besides other attacks, power analysis attacks are also a threat to smart cards. The taxonomy of attacks attempted on smart cards has been summarized in Figure 2. Physical attacks attempted on smart cards can be refuted using techniques such as MAC (message authentication code), cryptographic fingerprinting, polarization of the antenna, spread spectrum technologies, and measures of physical security. A resilient and efficient implementation of security schemes and protocols is needed to defeat information security attacks on smart cards.

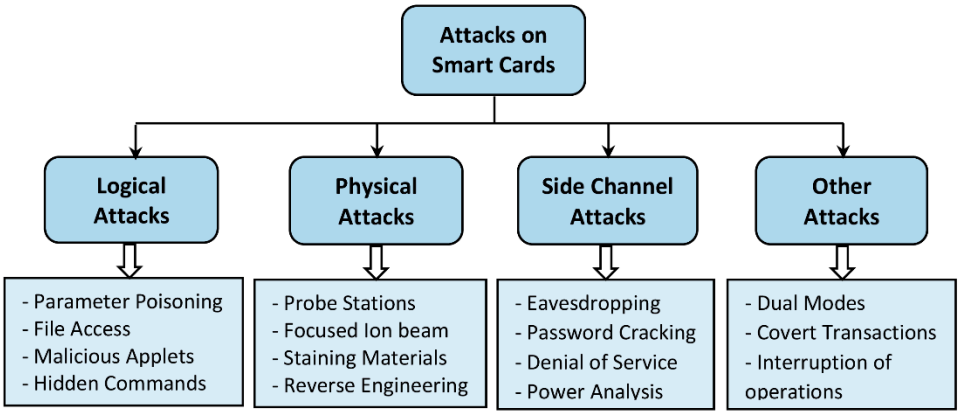

Figure 2. Taxonomy of attacks on smart cards.

1.3. Communication Scenario

With the evolution of new paradigms in computing, including cloud computing and internet of things (IoT), the private information of users is transmitted through the open public communication network. Therefore, it is important to ensure that the user's confidential information is protected from unauthorized disclosure. Due to this reason, a user as well as the server must verify the validity of each other before any data transmission happens. Generally, smart cards are used in a user-server environment where a legitimate user wishes to access the facilities offered by a server. A user, along with their smart card, has to authenticate themself with the server to access its services. Furthermore, the user has to confirm that they are interacting with a legitimate server and not with an adversary. This communication model has been demonstrated in Figure 3.

1.4. Organization of Paper

The remaining paper has been structured into seven different sections. Section 2 discusses the related work and briefly explains the latest and most efficient existing smart card authentication protocols. Section 3 focuses on the preliminaries used in developing the smart card security protocol. In the Section 4 of the paper, an elliptic curve signcryption based mutual authentication protocol for smart cards has been developed. The correctness of the proposed smart card authentication protocol has been verified in the Section 5 of the paper. The analysis of security properties, security functionalities, and performance measurement of the proposed protocol are discussed in Sections 6 and 7. Section 8

explains the results obtained from the analysis of security properties and performance of the proposed protocol, which is then followed by a conclusion in Section 9 of the paper.

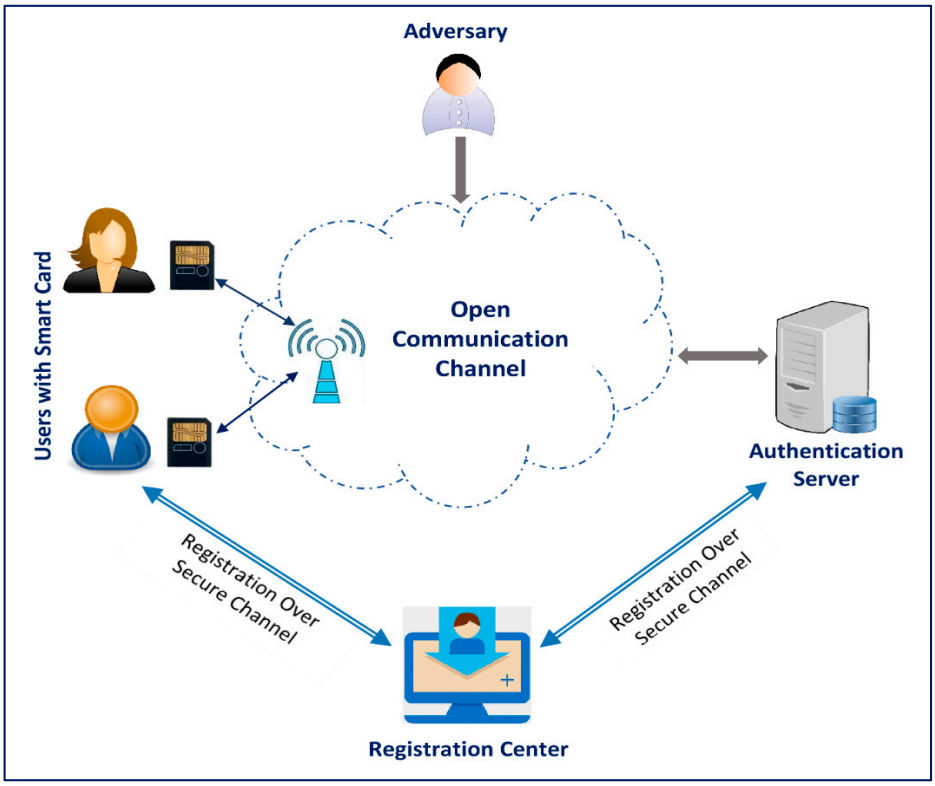

Figure 3. Communication model.

2. Literature Review

Authentication and key exchange are the primary security requirements in a user-server model. Zhao et al. [7] mentioned that, to implement the authentication process in a client-server environment, approaches based on the certificate, approaches based on the password, and approaches based on the identity are three possible methods. Among these three approaches, password-based approaches are found to be the most suitable in this environment, as they are comparatively simple and easy to implement. In password-based schemes, a user wanting to access some service selects a password and enters this password when using the services offered by a server. The server, in turn, maintains a table with the entries of identities and passwords of the users, so that the validity of passwords entered by the users can be verified. Two-tier authentication methods have been designed to overcome the problem of offline password guessing in password-based methods, as it requires the user to show the credentials through smart cards and enter the password. These two-tier authentication schemes have been found stronger and secure, because even if either of the two secret components, i.e., either the smart card or the password, are revealed to an opponent, the scheme remains secure. However, both components must not be exposed to the opponent together. Due to the higher security of two-tier authentication schemes, the authentication schemes utilizing the smart card and the password have both been discussed under the related work.

Chang and Wu [8] designed a two-tier authentication approach using both the smart card and the password, which enhanced the security level of password-based schemes. Das et al. [9] developed the concept of the smart card and password-integrated authentication mechanism based on dynamic identity. The advantage of the dynamic identity-based approach over the static identity-based approach is that the privacy of the user is not revealed when dynamic identity is used. However, Liao et al. [10] have explored the weakness in Das et al.'s [9] approach and proved that it is defenseless against

impersonation attack. They have also developed an integrated smart card and password authentication mechanism based on dynamic identity, which can counter the impersonation attack. Yeh et al. [11] and Khan et al. [12] have also presented improvements in the earlier presented schemes, but both these schemes have failed to provide security from offline password guessing attacks in the event of a smart card being lost.

Liao and Wang [13] have designed a dynamic identity-based authentication mechanism, to facilitate the user authentication in a multi-server environment, and it can be applied in multi-server settings. Hsiang and Shih [14] have revealed that the mechanism proposed by Liao and Wang [13] cannot counter an insider attack, forgery, or an impersonation attack. Hsiang and Shih have also presented an authentication approach which removed the weaknesses in the mechanism of Liao and Wang [13]. Later, Sood et al. [15] demonstrated that the scheme of Hsiang and Shih [14] is susceptible to replay attacks, impersonation attacks, and stolen smart card attacks. However, the protocol of Sood et al. [15] has been unable to defend smart cards from replay attacks and impersonation attacks.

Pippal et al. [16] have developed a multi-server authentication mechanism which does not use the verification table. This mechanism also facilitates users at remote location to access the services of multiple servers without individual registration. Furthermore, a user is able to change its password securely without registering with the registration center or the server. Later, Yeh [17] performed a critical analysis over Pippal et al.'s [16] mechanism and figured out that this mechanism failed to provide security from server counterfeit attacks, man-in-the-middle attacks, and user impersonation attacks. He also presented an enhanced scheme, which provides more security than Pippal et al.'s [16] mechanism.

Zhang et al. [18] developed a password-centered key agreement and authentication mechanism and announced that it is safe from all the security attacks. However, Farash and Attari [19] proved that Zhang et al.'s [18] scheme failed to resist impersonation and password changing attacks. They also designed a password-based session initiation mechanism using smart cards, but this scheme was not found to be secure from password guessing attacks. Odelu et al. [20] figured out that Islam's [21] scheme suffered from time synchronization problems. They developed and proposed an improved scheme, which overcame the weaknesses in Islam's [21] scheme. Wang et al. [22] presented a two-factor authentication mechanism, which provides user anonymity and untraceability while resisting de-synchronization attacks. They also claimed that their scheme did not incur any additional cost. However, the major weakness of Wang's scheme was that it failed to satisfy forward security.

Challa et al. [23] designed a three-factor scheme for key agreement schemes and authentication with session initiation based on ECC; however, it was found defenseless from replay attacks, password guessing attacks, and stolen smart card attacks [24]. Reddy et al. [25] projected an ECC-based authentication protocol with anonymity for mobile computing environment and announced that their protocol was efficient and secure. Later, Wu et al. [26] showed that Reddy et al.'s [25] protocol was unsuccessful in achieving mutual authentication and key agreement. In addition, it was susceptible to known session-specific temporary information attacks.

Chaudhry et al. [27] performed a comprehensive analysis of the authentication scheme presented by Huang et al. [28] and figured out that Huang et al.'s [28] scheme was prone to forgery and impersonation attacks. Furthermore, they explored that it also has correctness issues. Chaudhry et al. also developed an enhanced scheme overcoming the drawbacks and weaknesses of Huang et al.'s scheme. Truong et al. [29] designed an authentication mechanism on the basis of ECC for the multi-server environment and claimed that their schemes provided necessary security functionalities. However, Zhao et al. [7] performed a critical analysis of Truong et al.'s [29] approach and showed that it failed to resist password guessing attacks and the impersonation attacks. They also proposed a smart card mutual authentication protocol on the basis of ECC, which removed the deficiencies in the protocol presented by Truong et al.

Chandrakar and Om [30] presented a two-factor scheme based on the Rabin Cryptosystem implementing an authentication mechanism and key agreement for remote users. However, this scheme

involved modular exponentiation operations and is not appropriate for low computationally capable devices and environments including the Internet of Things. Jiang et al. [31] presented a three-factor mechanism for wireless sensor networks integrated with Internet which facilitates authentication and key setting. However, this scheme was not found to be efficient and consumed more computational time since it involved modular squaring and square root modulo operations. Qiu et al. [32] developed a mutual authentication mechanism for healthcare information systems based on ECC and announced that their scheme was efficient and could resist all kinds of security attacks. However, Zhang et al. [33] proved that the scheme proposed by Qiu et al. [32] was defenseless against insider attacks and denial of service attacks.

Kumari et al. [34] designed an authentication mechanism with session initiation using ECC and explained that their scheme was forward secure and could counter all kinds of attacks. However, Qiu et al. [35] showed that Kumari et al.'s [34] scheme was unsuccessful in offering forward security and could not protect against key-compromise and impersonation attacks. Limbasiya et al. [36] developed a smart card authentication approach for network applicants and showed that it could withstand known attacks and was more time efficient. However, Dharminder et al. [37] established that Limbasiya et al. 's [36] scheme was not secure against password guessing attacks, stolen card attacks, and disclosure of a session key. Sureshkumar et al. [38] presented an enhanced mutual authentication mechanism for session initiation and announced that it provided user anonymity. Sourav et al. [39] have shown that the approach presented by Sureshkumar et al. [38] does not protect the privacy of the user credentials and these can be revealed to an attacker.

Qiu et al. [40] announced an elliptic curve-based smart card security protocol and declared that their protocol was more secure and computationally efficient than the other existing ones. However, in a recent article, Nikooghadam and Amintoosi [41] proved that Qui et al.'s [40] protocol fails to offer mutual authentication and is vulnerable to Denial of Service attacks. Nikooghadam and Amintoosi [41] have also suggested an ECC smart card security protocol, which satisfies two-factor mutual authentication at the same time offering key agreement, and showed that this protocol can successfully resist all the attacks. However, Shouqi et al. [42] proved that the scheme of Nikooghadam and Amintoosi [41] fails to provide many security functions and cannot defend against many attacks. In addition, the protocol of Nikooghadam and Amintoosi [41] consumes more computational time than Zhao et al.'s [7] protocol.

Zhao et al. [43] have suggested a three-factor authentication mechanism based on Chebyshev chaotic maps for remote users and proved that their scheme can resist against all well-defined attacks, including impersonation attacks and offline password guessing attacks. However, Dharminder and Gupta [44] have demonstrated that chaotic maps-based security schemes fail to protect from stolen smart card attacks, identity guessing attacks, password guessing attacks, impersonation attacks, and session key attacks. Although chaotic maps-based security solutions consume average computational time, they fail to satisfy many security attributes, and hence, are not suitable for securing less computationally capable devices. Zheng et al. [45] have designed a mutual authentication protocol using elliptic curves for RFID systems, which provides many security features but does not use a password. In addition, the registration phase has not been mentioned in the work so that it can be used in the multi-server environment.

Wang et al. [46] proposed a three-factor authentication mechanism for cloud-based IoT, which satisfies a number of security properties and can counter many attacks, including stolen verifier attacks, forgery attacks, and de-synchronization attacks. However, the scheme of Wang et al. [46] is based on chaotic maps. Gzaffar et al. [47] presented an authentication scheme for remote data access over cloud-based systems, which provides proxy re-encryption, where the proxy re-encryption key is used by the cloud. This scheme successfully satisfies many security features and can counter attacks on confidentiality, access control, user anonymity, password guessing, device stolen, and non-repudiation. However, this scheme is based on pairing-based cryptography, which generates large size parameters and is not suitable for resource-constrained environments, including smart cards and IoT.

Yu et al. [48] have developed a mutual authentication scheme for cloud and IoT, which also provides key agreement. This scheme satisfies all the necessary security functionalities but uses bilinear pairings operations, which consume huge amounts of time as compared to elliptic curve-based schemes. Ostad-Sharif et al. [49] presented a key generation and mutual authentication mechanism for healthcare based on elliptic curve schemes and claimed that it could resist all kinds of attacks. However, Kumari et al. [50] performed a critical analysis of Ostad-Sharif et al.'s [49] scheme and proved that it was defenseless against password guessing attacks, key compromise, and impersonation attacks. Choudhary et al. [51] proposed a session key exchange and authentication mechanism for industrial IoT, which focuses on removing the security weaknesses of the existing schemes. This scheme utilizes both symmetric key and asymmetric key cryptography and can defend against modification attacks, replay attacks, and man-in-the-middle attacks. Recently, Mandal et al. [52] developed a three-factor signcryption-based access control approach for IoT surroundings. Although their scheme satisfies many security functions, including mutual authentication, and access control, and can resist many attacks, it consumes a total of 14 elliptic curve point multiplications, leading to a very high computational cost.

Rajasekar et al. [53] have presented a multi-factor authentication scheme including a password, user biometrics, and a smart card based on hyper elliptic curve signcryption and showed that it is safe from dictionary attacks, password guessing attacks, biometric recognition errors, denial of service attacks, and impersonation attacks.

In the present scenario, Java Cards have become popular in implementing the smart card security schemes. Java Card was developed with an objective to easily implement the security functionalities in the smart card using java language [54]. The very first Java Card was developed in 1996, and since then, many versions of Java Cards have been developed. Typically, a Java Card contains an 8- or 16-bit CPU of 3.7 MHz and RAM of 1 KB. It is significant to note that Java Cards have the capability to implement ECC-based algorithms, including the Elliptic Curve Diffie Hellman Algorithm (ECDHA) and Elliptic Curve Digital Signature Algorithm (ECDSA). Recently, Java Card 3.1 has been launched by the Sun Microsystems and intends to boost the security of IoT devices.

The analysis of related works shows that many elliptic curves-based smart card security protocols have been developed recently due to the efficiency of ECC based systems. Among the schemes discussed in the literature review, many schemes have been either unable to satisfy the security attributes required by the smart card systems or have failed in protecting against the attacks on smart cards. In addition to the schemes of Zhao et al. [7] and Chaudhary et al. [27], the protocol proposed by Mo et al. [55] also satisfies all the required security properties and can resist all kinds of attacks, but these three schemes have a high computational cost or enhanced communication overhead. Since the smart cards have limited processing ability, less memory storage availability, and restricted bandwidth, the security schemes for smart card-based systems must have little computational cost and communication bandwidth. In order to further reduce the computational time and bandwidth requirement, it will be better to combine the elliptic curve cryptosystem and signcryption and then apply it to secure less computationally capable environments such as smart cards.

3. Preliminaries

In this section, a brief overview of operations on elliptic curve, computational problems defined on elliptic curve, and introduction of signcryption have been discussed. The notation { } represents a set of values in further discussion.

3.1. Operations on Elliptic Curve

An elliptic curve E defined over finite field F_q is expressed by the famous Weierstrass Equation:

$$y^2 = x^3 + Ax + B \quad (1)$$

where $A,\ B \in F_q$ are constants that satisfy the equation:

$$4A^3 + 27B^2 \neq 0 \tag{2}$$

The points on the elliptic curve E are given by:

$$E(Fq) = \{O\} \cup \left\{(x,\ y) \in F_q \times F_q\ :\ y^2 = x^3 + Ax + B\right\} \tag{3}$$

where O is the point at infinity. The properties and operations satisfied by $E\left(F_q\right)$ are listed below.

- Identity—For each point existing on the elliptic curve E, i.e., $Q \in E\left(F_q\right)$, $Q + O = O + Q = Q$.
- Negative—Assuming a point $Q = (x,\ y \in E\left(F_q\right)$, then the negative of point Q is represented by $-Q = (x, -y)$. The property is $Q + (-Q) = O$. Moreover, $-O = O$.
- Elliptic Curve Point Addition—Assuming the two points $Q = (x_1,\ y_1)$ and $R = (x_2,\ y_2)$ on $E\left(F_q\right)$ with $Q \neq \pm R$. The elliptic curve point addition of Q and R is expressed by another third point $Q + R = (x, y) \in E\left(F_q\right)$. The coordinates x and y are given by:

$$x = \rho^2 - x_1 - x_2 \text{ and } y = \lambda(x_1 - x) - y_1 \tag{4}$$

where:

$$\rho = \frac{y_2 - y_1}{x_2 - x_1}, \text{ if } Q \neq R \text{ and } = \frac{3{x_1}^2 + A}{2y_1}, \text{ if } Q = R \tag{5}$$

- Elliptic Curve Point Multiplication—Consider a point $Q \in E\left(F_q\right)$, the elliptic curve point multiplication is expressed as $kQ = Q + Q + \ldots + Q(k\ times)$.
- XOR Operation—The XOR operation between two points on the elliptic curve has been computed by performing XOR operation between the corresponding x and y coordinates of both the points.

3.2. Computational Problems on Elliptic Curve

The elliptic curve cryptographic system is considered to be secure due to the existence of three computationally hard problems, which are listed below:

- Elliptic Curve Discrete Logarithmic Problem—For two given points on the elliptic curves $Q' \in E\left(F_q\right)$ and $R' \in E\left(F_q\right)$, it is computationally not feasible to find an integer k such that $Q' = kR'$ [56].
- Elliptic Curve Diffie-Hellman Problem—For a given point $Q' \in E\left(F_q\right)$ and two more points, $R' = aQ$ and $S' = bQ$, on the same elliptic curve $E\left(F_q\right)$, it is computationally hard to determine a point $P' = abQ'$ [57].
- Elliptic Curve Decision Diffie-Hellman Problem—For a given point $Q' \in E\left(F_q\right)$ and three more points, $R' = aQ'$, $S' = bQ'$, and $T' = cQ'$, on the same elliptic curve $E\left(F_q\right)$, it is computationally not feasible to conclude whether $T' = abQ'$ [58].

3.3. Elliptic Curve Signcryption

Signcryption is a well-established cryptographic technique which logically enables the functionality of authentication and confidentiality in a single step. The technique of signcryption becomes more efficient when combined with elliptic curve cryptography. Zheng and Imai [59] have shown that it saves 58% time in computation and 40% overhead in communication, when compared with the traditional signature-then-encryption approach, when used along with elliptic curve cryptography. Moreover, the combined functionality of signcryption and elliptic curve cryptography can satisfy many security attributes, at the same time resisting many attacks. Due to this, it will be wise to use elliptic curve-based signcryption for securing a resource constraint device such as a smart card.

In order to provide a preview of how the elliptic curve based signcryption scheme can be developed, a brief overview of the very first elliptic curve signcryption scheme developed by Zheng and Imai [59] has been presented here. The two variants of Shortened Elliptic Curve Digital Signature Scheme (SEDSS) used in the elliptic curve signcryption scheme of Zheng and Imai have been shown in Table 1. The three phases of this approach are the initialization phase, the signcryption phase, and the unsigncryption phase.

Table 1. Variants of elliptic curve digital signature scheme.

Shortened Scheme	Signature {t,w}	Signature Verification
Scheme 1	$t = Hash\ (vG, M)$ $w = (v/t + x) mod q$	$Z = w(X + tG)$ $Check\ if\ Hash(Z, M) = t$
Scheme 2	$t = Hash\ (vG, M)$ $w = (v/1 + xt) mod q$	$Z = w(G + tX)$ $Check\ if\ Hash(Z, M) = t$

q: A large prime number G: Base point on elliptic curve v: Random number chosen from $[1 \ldots q-1]$ x: Private key of sender X: Public key of sender ($X = xG$).

3.3.1. Initialization Phase

In this preliminary phase, the global public parameters are chosen. In addition, the key pairs for the sender and the receiver are also generated. The steps carried out in the initialization phase are explained below:

- Alice, the sender, and Bob, the receiver, agree on the elliptic curve E, which is defined over finite field $GF(q^n)$ with $q \geq 2^{160}$ and $n = 1$, or $q = 2$ and $n \geq 160$.
- A big prime p with order q^{m-1} is selected.
- G of order p on elliptic curve E is randomly chosen, which becomes the base point of the elliptic curve.
- Keyed hash function KH is chosen.
- Hash function H is selected.
- Encryption algorithm ENC and decryption algorithm DEC are agreed upon.
- Alice randomly selects a private number v_x in the range $[1 \ldots p-1]$, which is her private key.
- Alice generates her own public key P_x by computing:

$$P_x = v_x G \tag{6}$$

- Bob randomly selects a private number v_y in the range $[1 \ldots p-1]$ which is his private key.
- Bob generates his own public key P_y by computing:

$$P_y = v_y G \tag{7}$$

3.3.2. Signcryption Phase

In the signcryption phase, the message is encrypted and digitally signed. The sender Alice wanting to send a message M, randomly selects an integer d from $[1 \ldots p-1]$ and performs the computations given below.

- Generate the key K as:

$$K = H(dP_y) \tag{8}$$

- Divide the key K is into two equal size subkeys k and k'.
- Encrypt the message M using subkey k as:

$$c = E_k(M) \tag{9}$$

- Compute t using subkey k' as $t = KH_{k'}(M,\ Blind\_Info)$
- Compute s as:

$$s = \frac{d}{t + v_x}\ mod\ q \tag{10}$$

- Then, Alice creates the signcrypted message $\{c, t, s\}$ and sends it to Bob.

3.3.3. Un-Signcryption Phase

In the unsigncryption phase, decryption and signature verification are performed at the receiving side. Upon receiving the signcrypted message $\{c,\ t,\ w\}$ from Alice, Bob performs the computations as given below.

- Compute $w = sv_y\ mod\ q$
- If Scheme 1 is used then it regenerates the key K as:

$$K = H(wP_x + wtG) \tag{11}$$

- If Scheme 2 is used then it regenerates the key K as:

$$K = H(wG + wtP_x) \tag{12}$$

- Decrypt the ciphertext as:

$$M = D_k(c) \tag{13}$$

- Accept the message M if:

$$KH_{k'}\left(M,\ Blind_{Info}\right) = t \tag{14}$$

where Scheme-1 is the Shortened Elliptic Curve Digital Signature Scheme 1 and Scheme-2 is the Shortened Elliptic Curve Digital Signature Scheme 2. In this way, the functionality of the digital signature and encryption is achieved in a single logical phase.

4. Proposed Elliptic Curve Signcryption Based Security Protocol for Smart Card (ECSSP-SC)

With the evolution of computing paradigms such as cloud computing, distributed computing, and edge computing, different services are provided by the enterprises simultaneously to the users. A secure and efficient mutual authentication scheme for multi-server environment is required to ensure the access of these services by the authorized users. In this segment of the paper, an Elliptic Curve Signcryption based Security Protocol for Smart Cards (ECSSP-SC) has been presented and explained. The main focus of the protocol presented here is to achieve mutual authentication in a multi-server setting.

Although the ECSSP-SC presented here has been developed for multi-server environment, its mutual authentication phase can also be used for M2M and U2U communication systems, with a slight change in the registration phase. In this case, the machine or user should register only with the server and the presence of a Registration Center can be removed.

The proposed protocol has been divided into a setup phase, registration phase, signcryption-based mutual authentication phase, and update phase for better understanding and explanation. The three entities participating in the proposed protocol are the user having a smart card, *USR*, the server providing services, *SVR*, and the registration center, *RGC*. The symbols and denotations utilized in the presentation and description of the proposed protocol have been listed in Table 2. The operations described in the Section 3.1 have been used in the explanation of the proposed ECSSP-SC.

Table 2. Symbols and notations used in the proposed Elliptic Curve Signcryption based Security Protocol for Smart Cards (ECSSP-SC).

Symbol	Denotation
q, n	Two large prime numbers
F_q	Finite field over prime number q
E	Elliptic curve defined on F_q
A, B	Parameters for elliptic curve E
O	Point on infinity
G	Base point of E
USR	User having smart card
SVR	Server providing service
RGC	Registration center
v	Randomly selected integer by USR
I_R	Identity of RGC
I_S	Identity of SVR
I_U	Identity of USR
$Hash$	Hash computation
P_S	Server's public key
PWD	Password of USR
T_C	Current timestamp
t	Expected delay

4.1. Setup Phase

In the setup phase, the registration center RGC chooses the common parameters for the system and publishes them globally. The steps in the setup phase are:

- RGC chooses an elliptic curve E defined on the finite field F_q having parameters $\{q, A, B, G, n\}$. For parameters A, B, the condition $4A^3 + 27B^2 \neq 0$ is satisfied.
- RGC also chooses a hash function $Hash: \{0,1\}^* \rightarrow \{0,1\}^l$.
- The registration center RGC publishes the global system parameters $\{q, A, B, G, n, Hash\}$.

4.2. Registration Phase

Both USR and the SVR register themselves with the registration center RGC in the registration phase of the ECSSP-SC. It is significant to note that all the steps in the process of registration are executed over a secure channel. Furthermore, it has been assumed that no communication can be intercepted by third party or intruder over the secure channel. There can be many forms of secure channels depending on the usage, including personal visits and courier service.

4.2.1. Registration of the SVR with RGC

The steps performed for the registration of the SVR with RGC are given below. A pictorial representation of this process has also been shown in Figure 4.

- The server SVR selects its identity I_S. It is assumed that the identity I_S of the server is kept secret from all the parties except RGC.
- SVR creates its public key as $P_S = I_S G$ and communicates the message $\{I_S\}$ to the RGC.
- After reception of the message $\{I_S\}$, the RGC chooses a key K_{RGC} and computes:

$$v_R = Hash(K_{RGC}) \tag{15}$$

RGC also generates the key:

$$K_{RS} = (v_R \oplus I_S \oplus I_R) \tag{16}$$

- RGC sends the message $\{K_{RS}\}$ to the SVR.

- *SVR* saves the key K_{RS} confidentially.

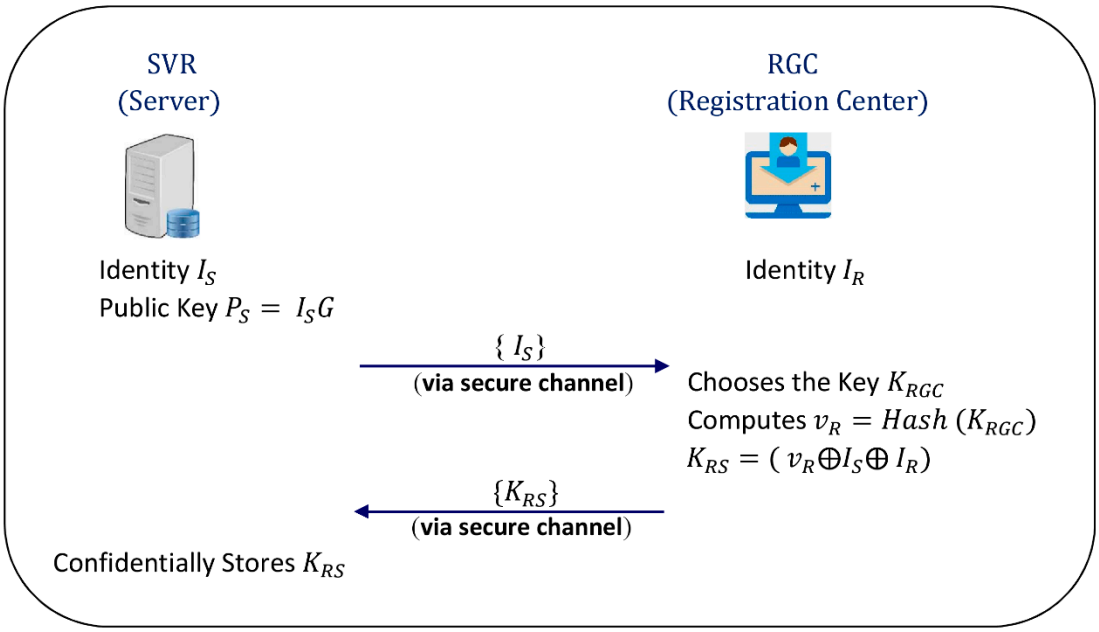

Figure 4. Registration of *SVR* with *RGC* in ECSSP-SC.

4.2.2. Registration of the *USR* with *RGC*

Similarly, the computation performed in the registration of the user *USR* with *RGC* is given below and the pictorial representation of this process has been demonstrated in Figure 5.

- The user *USR* chooses its identity as I_U and sets the password as *PWD*. The *USR* computes:

$$S_U = I_U G \tag{17}$$

and saves S_U.

- *USR* randomly selects $x \in Z_n$ and computes its public key as:

$$P_U = xG \tag{18}$$

The user also computes:

$$R_U = (PWD \oplus P_U) \tag{19}$$

and directs the message $\{I_U, R_U\}$ to the *RGC*.

- On getting the message $\{I_U, R_U\}$, the *RGC* generates the key:

$$K_{RS}' = (v_R \oplus I_R) \tag{20}$$

and computes:

$$A_{RU} = (K_{RS}' \oplus R_U) \tag{21}$$

RGC issues a smart card with $\{A_{RU}, Hash(), G\}$ to the *USR*.

- On getting the smart card, the *USR* saves x on the smart card.

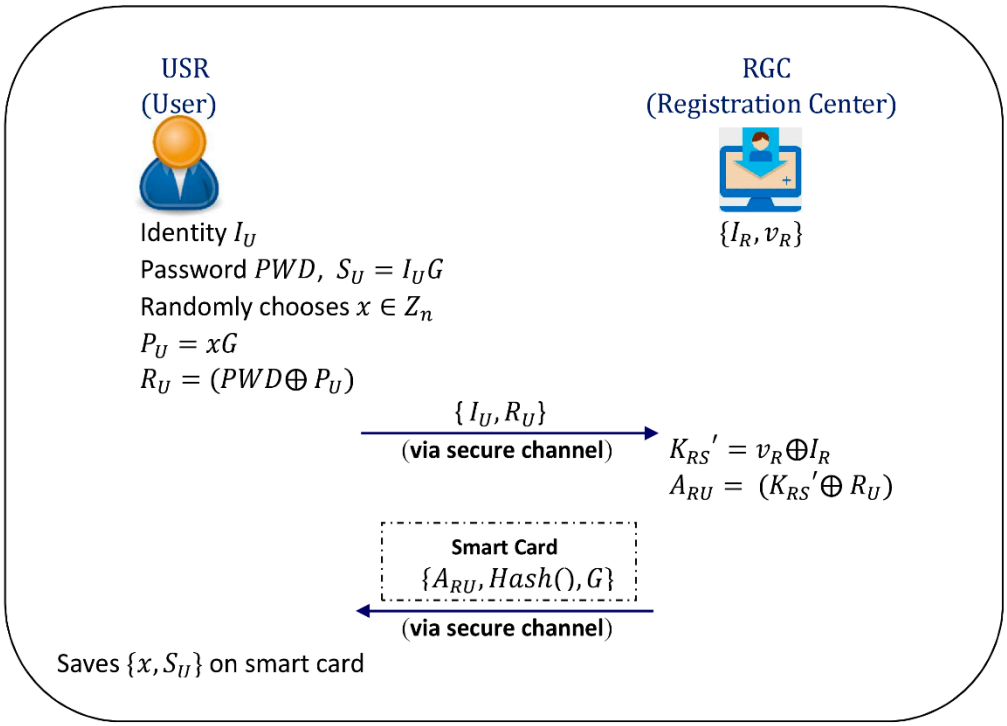

Figure 5. Registration of *USR* with *RGC* in ECSSP-SC.

4.3. Signcryption Based Mutual Authentication Phase in ECSSP-SC

In the signcryption-based mutual authentication phase of the proposed ECSSP-SC, both *USR* and *SVR* authenticate each other. It has been assumed that this process of mutual authentication occurs over the open communication channel. The *USR* ensures that the user is utilizing the services of a genuine server, while the *SVR* checks that it is interacting with a valid user. This phase has been implemented by utilizing the security and performance benefits of elliptic curve-based signcryption. The algorithm of the elliptic curve signcryption-based mutual authentication phase in the proposed ECSSP-SC has been shown in Figure 6 and the steps executed in this phase are explained below.

- The *USR* having the private public key pair $\{x, P_U\}$ first randomly chooses an integer v and inserts the smart card into the card reader/device to start the authentication process. After entering its identity I_U and the password PWD, the user *USR* does the following computations:

  Calculate:
  $$V = vP_S \tag{22}$$
  Generate the key:
  $$K = (V \oplus A_{RU} \oplus R_U) \tag{23}$$
  Create the ciphertext:
  $$c = E_K(S_U) \tag{24}$$
  Compute:
  $$r = Hash(c \oplus K) \tag{25}$$
  Compute:
  $$w = v/r \tag{26}$$
  Calculate:
  $$T = rG \tag{27}$$

The user *USR* records the timestamp T_U and transmits the message $\{c, T, w, T_U\}$ to the *SVR*. The components $\{c, T, w\}$ produced in this step implement the functionality of signcryption. Here, the component c fulfills the functionality of confidentiality by encrypting the P_U, the component T is used to carry out the authentication functionality, and the component w facilitates the masking of secret values.

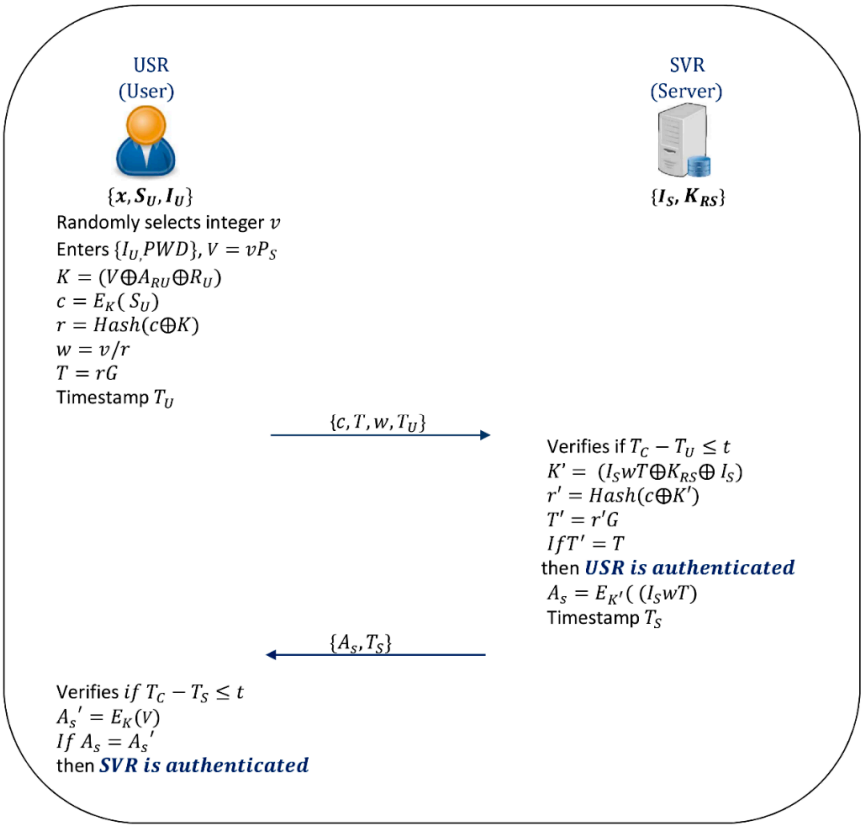

Figure 6. Signcryption-based Mutual Authentication Phase in ECSSP-SC.

- Upon receiving the message $\{c, T, w, T_U\}$ from the *USR*, the *SVR* performs the calculations given below:

  Confirm whether:

$$T_C - T_U \leq t \tag{28}$$

where T_C is the present time and t is the average delay in transmission. If $T_C - T_U > t$, then the *SVR* terminates the session. Otherwise, for the *SVR*:

Retrieve the key as:

$$K' = (I_S wT \oplus K_{RS} \oplus I_S) \tag{29}$$

Compute:

$$r' = Hash(c \oplus K') \tag{30}$$

Compute:

$$T' = r'G \tag{31}$$

If $T' = T$, then the *SVR* successfully authenticates the *USR*. In this way, the functionality of un-signcryption is fulfilled in which T' is generated by the *SVR* and matched with the received T.

Moreover, *SVR* computes:

$$A_s = E_{K'}(I_S wT), \tag{32}$$

which then records timestamp T_S and transmits the message $\{A_s,\ T_S\}$ to the *USR*.

- On receipt of the message $\{A_s,\ T_S\}$ from the SVR, the USR does the following:

  Verify whether:

$$T_C - T_S \leq t \tag{33}$$

  If $T_C - T_S > t$, then the session is dismissed. Otherwise, the USR computes:

$$A_s{}' = E_K(V) \tag{34}$$

  If $A_s = A_s{}'$, then the USR successfully authenticates the SVR.

4.4. Password Update Phase

If the USR wishes to update or change their password as PWD^{New}, the following steps are followed:

- The USR selects a service provider for changing their password.
- The USR and the service provider are first authenticated mutually. After successful authentication, the USR updates their password as PWD^{New}.
- The USR makes the smart card compute:

$$A_{RU}{}^{New} = \left(A_{RU} \oplus PWD \oplus PWD^{New}\right) \tag{35}$$

- The smart card replaces A_{RU} with $A_{RU}{}^{New}$ in its storage.

5. Proof of Correctness

The correctness and accuracy of the mutual authentication phase of the proposed ECSSP-SC can be confirmed by analyzing the keys K and K' created by the USR and the SVR, respectively. If the key K of the USR and key K' of the SVR is same, then the proposed protocol correctly performs the mutual authentication process.

- The key K generated by the USR is given by:

$$\begin{aligned} K &= (V \oplus A_{RU} \oplus R_U) = vP_S \oplus A_{RU} \oplus R_U \\ &= (v\, I_S G \oplus K_{RS}{}' \oplus R_U \oplus R_U) \\ &= (v\, I_S G \oplus K_{RS}{}') = (v\, I_S G \oplus v_R \oplus I_R) \end{aligned} \tag{36}$$

  since $P_S = I_S G$, $A_{RU} = K_{RS}{}' \oplus I_U \oplus R_U$, and $K_{RS}{}' = (v_R \oplus I_R)$.
- The key K' generated by the SVR is given by:

$$\begin{aligned} K' &= (I_S wT \oplus K_{RS} \oplus I_S) = (I_S \tfrac{v}{r} rG \oplus v_R \oplus I_S \oplus I_R \oplus I_S) \\ &= (v\, I_S G \oplus v_R \oplus I_R) \\ &= K \text{ which is the key generated by } USR \end{aligned} \tag{37}$$

  since $K_{RS} = (v_R \oplus I_S \oplus I_R)$.

  Therefore, the correctness of the ECSSP-SC is verified.

6. Analysis of Security Functionalities of ECSSP-SC

The security functionalities of the presented ECSSP-SC can be divided into two logical parts—security properties fulfilled by the proposed protocols and the resistance capability of the proposed protocol from different attacks.

6.1. Analysis of Security Attributes of ECSSP-SC

It has been mentioned in Section 1 that the required security attributes for a smart card-based system are integrity, authentication, forward security, confidentiality, availability, and non-repudiation. In this sub-section, the detailed analysis of security properties fulfilled by the ECSSP-SC has been carried out. A few reasonable assumptions, given below, have been made to justify this security analysis.

Assumption 1. *The secret v and the password PDW of the USR is kept secure and an Attacker A cannot access it. The identity I_S of the server is also secret.*

Assumption 2. *An Attacker A can obtain common system parameters.*

Assumption 3. *An Attacker A is unable to break the ciphertext generated by the encryption algorithm E_K.*

Assumption 4. *An Attacker A cannot obtain x in $P_U = xG$, given P_U and G because of the security of ECDLP.*

Assumption 5. *An Attacker A is unable to get x and A_{RU} from the smart card.*

6.1.1. Confidentiality

This attribute ensures that the message cannot be understood by the *Attacker A*. In the mutual authentication step of the proposed protocol, two messages, $\{c, T, w, T_U\}$ and $\{A_S.T_S\}$, are transmitted by the *USR* and *SVR*, respectively. In the first message $\{c, T, w, T_U\}$, an *Attacker A* cannot break the ciphertext c as per Assumption 3. According to Assumption 4, the *Attacker A* cannot deduce r from T since $T = rG$. Moreover, an attacker would fail to deduce any secret form w, since it is the division of v and r. In the second message $\{A_S.T_S\}$, according to the Assumption 3, A_S cannot be broken by the *Adversary A* and the key K' cannot be obtained. The second and the last quantity is the timestamp T_S. It may be concluded that from both the messages $\{c, T, w, T_U\}$ and $\{A_S.T_S\}$, no secret can be revealed by an *Attacker A*, and therefore, the proposed ECSSP-SC satisfies the confidentiality property.

6.1.2. Mutual Authentication

- *Authentication of the USR by the SVR*—When the *SVR* gets the signcrypted message $\{c, T, w, T_U\}$ from the *USR*, it first verifies the timestamp and regenerates the key K' by using I_S and K_{RS}. Then, it computes r' and T', and if $T' = T$, the *USR* is authenticated. The value of T generated by the *USR* depends on r and G, since $T = rG$. The value of r depends on the ciphertext c and the key K. The key K generated by the *USR* is a function of secret v and password *PWD*. If an *Attacker A* tries to pretend to be a genuine user, then they must produce the correct value of T. However, as per Assumption 1, the secret v and password *PWD* cannot be obtained by *Attacker A*. Thus, the signature of the user *USR* is unforgeable.
- *Authentication of the SVR by the USR*—On receiving the message $\{A_S.T_S\}$ from the *SVR*, the *USR* computes $A_s' = E_K(V)$, and if $A_S' = A_S$, it authenticates *SVR*. Since $A_s = E_{K'}(I_S w T)$, i.e., encryption of the secret $I_S w T$, it is available only with the *SVR* and due to the strength of the encryption algorithm mentioned in Assumption 3, an Attacker *A* can not reveal any secret form A_s and cannot forge it.

Thus, ECSSP-SC successfully provides mutual authentication between *USR* and *SVR*.

6.1.3. Integrity

If an *Attacker A* somehow alters any component c, T, or w in the message sent by the *USR*, then this modification will be easily identified by the *SVR*, since the key K' retrieved by the *SVR* will not be same as the key K generated by the *USR*. In turn, $T' \neq T$, and the session is terminated by the *SVR*, since the authentication fails. Similarly, when an *Attacker A* alters the component A_S of the message sent by the *SVR*, it will be easily detected by the *USR*, since A_S' computed by the *USR* will be different from A_S and the authentication of *SVR* will fail. Therefore, the current session will be dismissed by *USR*. Thus,

an attacker cannot fool the *USR* or *SVR* after modifying the messages, and hence, the ECSSP-SC also fulfills the integrity of messages.

6.1.4. Forward Security

Even if an *Attacker A* accidentally obtains the password *PWD* of the user, it cannot deduce the past messages $\{c, T, w, T_U\}$ and $\{A_S.T_S\}$, since these messages are based on the keys K and K', respectively, which in turn are a function of the private random number v selected by the server. In this way, the proposed ECSSP-SC is forward secure, since the past messages cannot be obtained by an *Attacker A*.

6.1.5. Availability

As per Assumption 1, the password of the *USR* is not exposed to the attacker. Furthermore, the password *PWD* of the *USR* is updated synchronously with the service provider and the smart card that is issued to the user. Except for this, there is no synchronous updating of information in the protocol. Thus, ECSSP-SC satisfies de-synchronization and provides availability.

6.1.6. Anonymity

The presented ECSSP-SC satisfies user anonymity, since the identity of the *USR* cannot be revealed from any of the two messages transmitted in the protocol. From any component of the first message $\{c, T, w, T_U\}$, the identity of the *USR* cannot be obtained because c is obtained by encrypting the P_U with key K, which is computed secretly. Moreover, according to Assumption 3, the attacker cannot break ciphertext c. As per Assumption 4, the attacker cannot deduce r from T since $T = rG$. Moreover, an attacker fails to deduce any secret from w since it is the division of v and r. Similarly, the attacker will fail to deduce any secret from the message $\{A_S.T_S\}$, since they need to break the cipher A_S, which is infeasible as per Assumption 3.

6.1.7. Non-Repudiation

The components c, T, and w of the message sent by *USR* depend upon the key K, which further depends on the private v and *PWD*. Similarly, the component A_S in the message sent by the *USR* also depends on the key K', which in turn, depends on the secret K_{RS}. Therefore, if $T' = T$, *USR* cannot repudiate that it transmitted $\{c, T, w, T_U\}$ and similarly, if $A_S = A_S{}'$, the SVR cannot repudiate that it has transmitted the message $\{A_S.T_S\}$, since the identity I_S of the server is kept secret. Therefore, ECSSP-SC successfully satisfies the non-repudiation attribute as well.

6.2. Analysis of Resistance Capability of ECSSP-SC against Attacks

Any secure protocol must have the capability to defeat attacks launched over it. In this sub-section, the resistance capability of the ECSSP-SC against different attacks has been analyzed. For modeling the active and the passive attacks attempted over the system, the attack model given by Ouafi and Phan [60] has been utilized, which has defined the three queries given below.

- $Q1$: *Execute* (USR, SVR, j): This query intends to model passive attacks where an *Attacker A* captures the truthful execution of session j between the *USR* and *SVR*.
- $Q2$: *Send* $(X, Y.j, MSG)$: This query intends to models active attacks where an *Attacker A* mimics the behavior of the party $X \in USR$ in the j^{th} session of the protocol, and transmits the message *MSG* to some $Y \in SVR$.
- $Q3$: *Corrupt*(USR, K): This query permits an *Attacker A* to acquire the secret key K stored with the *USR*.

6.2.1. Resistance to Replay Attacks

An *Attacker A* resides between the *USR* and *SVR* to record the messages $\{c, T, w, T_U\}$ and $\{A_S.T_S\}$ in a replay attack. The attacker then replays these messages later to the *SVR* or *USR*, respectively, to yield

an illicit effect. In the ECSSP-SC, the current timestamp has to be provided in both the messages. Thus, if an *Attacker A* tries to replay the past recorded message $\{c, T, w, T_U\}$ to the *SVR*, then, $T_C - T_U > t$ and the *SVR* will reject it. Similarly, if an *Attacker A* tries to replay the past recorded message $\{A_S.T_S\}$ to the *USR*, then $T_C - T_S > t$, and the *USR* will reject it. In addition, the fresh private random number v is used by the *USR* in the computation of the key K, and if the attacker replays the recorded message to fool the *SVR* or *USR*, it will be detected easily, since the key K or K' generated will be different and the authentication will fail. In this way, the proposed ECSSP-SC offers security from replay attacks.

6.2.2. Resistance to User Impersonation

For impersonating the *USR*, an *Attacker A* must be capable of generating the correct message $\{c, T, w, T_U\}$. The value of the components c, T, and w depends on the key K, which in turn depends on the secret v and A_{RU}, and as per the Assumption 5, an *Attacker A* cannot obtain them to create the key K and the correct message $\{c, T, w, T_U\}$. Therefore, the ECSSP-SC is resistant against user impersonation. The process of resisting user impersonation by ECSSP-SC has been demonstrated in Figure 7.

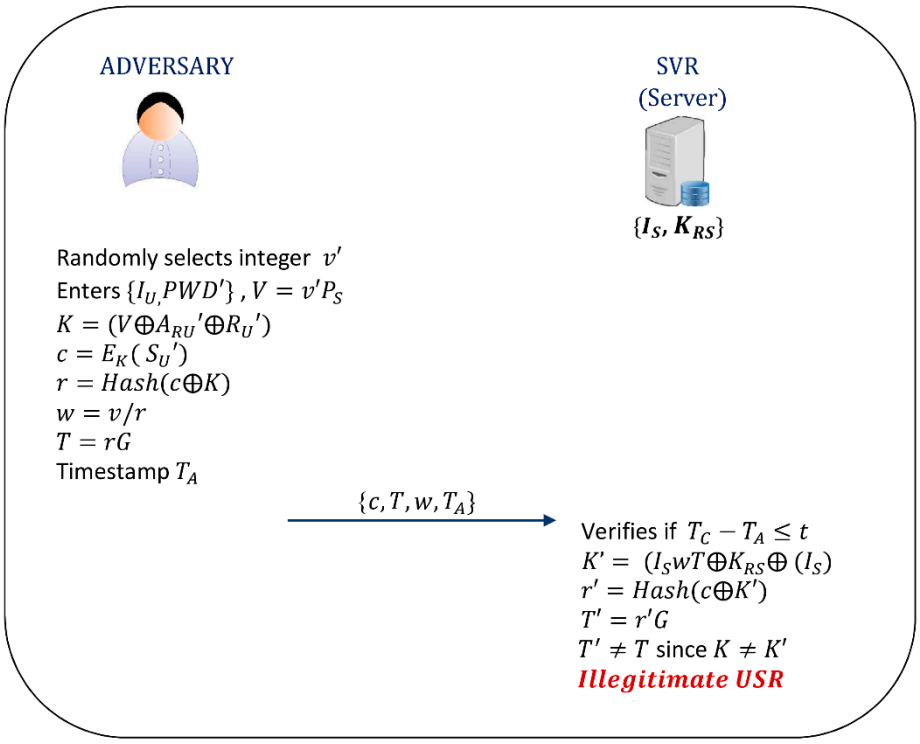

Figure 7. Resistance to user impersonation in ECSSP-SC.

6.2.3. Resistance to Server Impersonation

If an *Attacker A* tries to impersonate the *SVR*, then it has to create the accurate message $\{A_s, T_S\}$. As mentioned in point (ii), the *Attacker A* is unable to obtain the key K and they cannot access the secret key K_{RS}. Therefore, the *Attacker A* fails to impersonate the *SVR*, and the ECSSP-SC is safe from server impersonation. Resistance to server impersonation by ECSSP-SC has been shown in Figure 8.

6.2.4. Resistance to Insider Attacks

During insider attacks, the corrupt insider party, i.e., *RGC* or *SVR*, tries to obtain the password *PWD* of the *USR*. The *USR* sends the message $\{I_U, R_U\}$ to the *RGC*. From R_U, the *RGC* is unable to extract the password since $R_U = (PWD \oplus P_U)$ and $P_U = vG$. *RGC* cannot obtain v, since it is privately generated by the *USR*. Hence, the *RGC* cannot obtain the password of the *USR*. Moreover,

the *SVR* cannot also attain the password *PWD* of the *USR* from the contents of the message $\{c, T, w, T_U\}$, since the anonymity has been preserved in this message. Therefore, the proposed ECSSP-SC successfully provides defense from the insider attack.

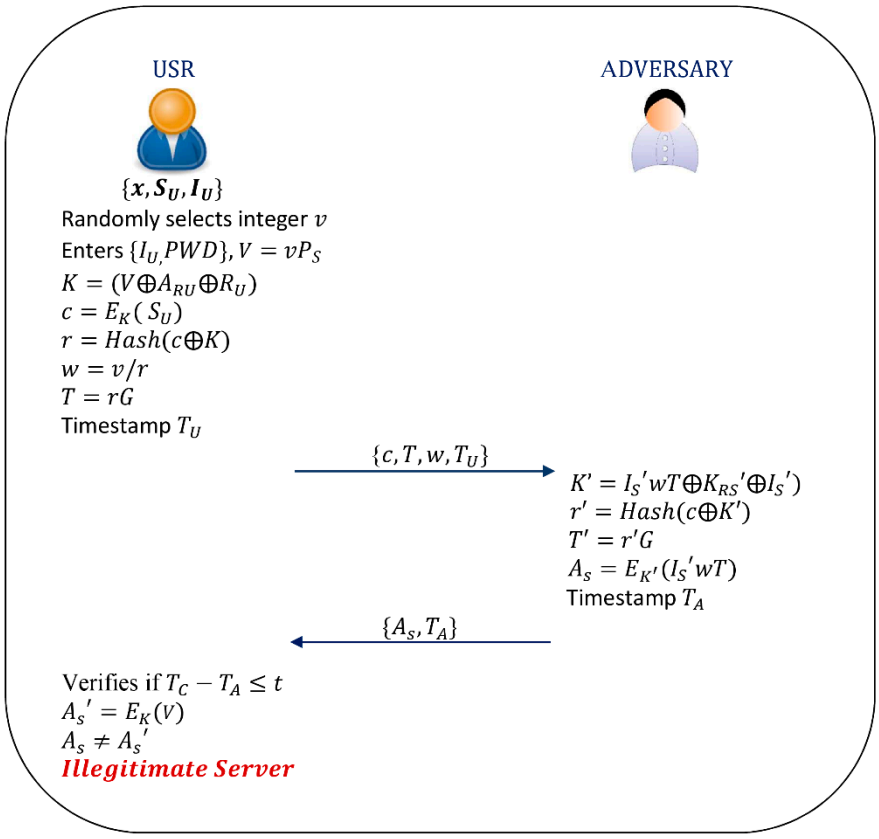

Figure 8. Resistance to server impersonation in ECSSP-SC.

6.2.5. Resistance to Offline Password Guessing Attacks

It has been explained and shown that no confidential information including the user's password *PWD* can be obtained from both the messages $\{c, T, w, T_U\}$ and $\{A_S.T_S\}$. Furthermore, any component from both the messages cannot be used in verifying the validity of the password guessed by an *Attacker A*. Furthermore, without the possession of secret v, it is infeasible for the *Attacker A* to confirm the validity of the guessed password. According to Assumption 1 and Assumption 5, secret v cannot be revealed. Thus, ECSSP-SC can successfully counter offline password guessing attacks.

6.2.6. Resistance to Known Key Attacks

An *Attacker A* somehow obtains the session key K and then tries to get the other session key during a known session key attack. In the ECSSP-SC, the session key K is generated by using secret random number v, which is selected fresh in every session. This ensures that all the session keys do not depend on each other. Thus, the proposed ECSSP-SC is safe from any known key attack.

6.2.7. Resistance to De-Synchronization Attacks

It has been mentioned that in ECSSP-SC, the password *PWD* of the *USR* is updated synchronously with the service provider and the smart card, and there is no other synchronous updating of any other information. Thus, ECSSP-SC also ensures resistance to de-synchronization attacks.

6.2.8. Resistance to MITM (Man-in-the-Middle) Attacks

In this attack, an *Attacker A* sits between the *USR* and the *SVR*. From there, it tries to mimic the behavior of either the *SVR* or the *USR*, i.e., it modifies the messages in transit to fool the other party. Since the ECSSP-SC can successfully resist user impersonation and server impersonation, it can also counter MITM attacks.

7. Performance Analysis of the Proposed ECSSP-SC

In this section, a comparison of the performance of the ECSSP-SC and other related protocols has been made concerning the computational cost consumed, the security attributes satisfied, and the resistance capability against attacks. This comparison has been presented to justify that the proposed ECSSP-SC is computationally more cost-efficient in comparison to the other related protocols, and at the same time, also satisfies all the essential security functionalities. Since the registration is a one-time activity, the computational time and communication overhead of the registration phase has been ignored in the analysis.

Before proceeding with the performance analysis, it is significant to compare the proposed ECSSP-SC with the related schemes with respect to the technique used for mutual authentication. Thus, a comparison of different smart card mutual authentication schemes based on the technique used are shown in Table 3.

7.1. Analysis of Computational Cost

The computational time of a security protocol can be analyzed by first counting the most time-consuming operations, and then multiplying them with the actual time consumed in completing the respective operations. Finally, the time consumed by each operation is added to find out the total time consumed by the protocol. In the analysis of the computational cost, only the time consumed in the authentication phase has been considered, since the registration of the *USR* and the *SVR* is a one-time process. Xie et al. [61] mentioned that by equipping a 2.5 GHz Intel i5 CPU holding 8 GB RAM, one ECPM operation consumes 2.501 ms and a single modular exponentiation consumes 3.043 ms. Since the time consumed in an ECPM operation and modular exponentiation operations is very high as compared to the other operations, the time invested in the operations other than ECPM and modular exponentiation has been ignored in the analysis. The count of ECPM and modular exponentiation operations in the smart card security protocols are shown in Table 4. A comparative analysis of the computational cost of the ECSSP-SC and related protocols was performed and the pictorial representation of the computational cost analysis of smart card security protocols are publicized in Figure 9.

Table 3. Technique used in different smart card authentication schemes.

Protocol	Technique Used		
	Modular Exponentiation	Elliptic Curve Cryptography	Signcryption
Zhao et al. [7]	×	✓	×
Yeh [11]	✓	×	×
Wang et al. [22]	✓	×	×
Chaudhary et al. [27]	×	✓	×
Truong et al. [29]	×	✓	×
Ghaffar et al. [47]	×	✓	×
Ostad-Sharif et al. [49]	×	✓	×
Mo et al. [55]	×	✓	×
Xie et al. [61]	×	✓	×
ECSSP-SC	×	✓	✓

✓—used, ×—not used.

7.2. Analysis of Communication Cost

Communication cost or overhead of ECSSP-SC has been calculated by computing the size of the messages transmitted between the *USR* and the *SVR* during the execution of the mutual authentication phase of the protocol. In the proposed ECSSP-SC, two messages are transmitted, $\{c, T, w\}$ and $\{A_S, T_S\}$. Assuming that 160-bit ECC parameters have been used, SHA-1 has been applied for producing the hash code, and AES-128, along with a 128-bit key, has been applied for encryption and decryption.

The communication overhead of the ECSSP-SC is computed as follows:

Overhead for the *USR*:

$$= 256 + 320 + 160 = 736 \text{ (bits)} \quad (38)$$

Overhead for the *SVR*:

$$= 256 + 160 = 416 \text{ (bits)} \quad (39)$$

Thus, the total communication overhead of the ECSSP-SC:

$$= 736 + 416 = 1152 \text{ (bits)} \quad (40)$$

A comparative analysis of the communication overhead of the ECSSP-SC and related protocols was performed and the graphical analysis of the comparison is shown in Figure 10.

7.3. Comparison of Security Attributes

It has been explained in the Section 6 that the proposed ECSSP-SC satisfies mutual authentication, the confidentiality of messages, the integrity of messages, non-repudiation, anonymity, availability, and forward security attributes. A comparison of the security properties of the ECSSP-SC and other protocols is shown in Table 5. From this analysis, one can conclude that the proposed mutual authentication protocol offers all the necessary security attributes of a smart card-based system.

Table 4. Number of operations in different smart card security protocols.

Protocol	No. of ECPM and Modular Exponentiation Operations Performed					
	USR		*SVR*		Total	
	P_M	M_E	P_M	M_E	P_M	M_E
Zhao et al. [7]	2	0	2	0	4	0
Yeh [11]	0	2	0	4	0	6
Wang et al. [22]	0	2	0	1	0	3
Chaudhary et al. [27]	3	0	3	0	6	0
Truong et al. [29]	2	0	2	0	4	0
Ghaffar et al. [47]	3	0	2	0	5	0
Ostad-Sharif et al. [49]	2	0	2	0	4	0
Mo et al. [55]	3	0	3	0	6	0
Xie et al. [61]	3	0	3	0	6	0
ECSSP-SC	2	0	2	0	4	0

P_M—elliptic curve point multiplication, M_E—modular exponentiation.

7.4. Comparison of Resistance Capability against Attacks

Section 6 has shown that the proposed mutual authentication protocol possesses the capability to counter replay attacks, user impersonation, server impersonation, insider attacks, known key attacks, password guessing attacks, de-synchronization attacks, and MITM attacks. A comparative analysis of the attack resistance ability of the ECSSP-SC and other protocols was performed and is shown in Table 6.

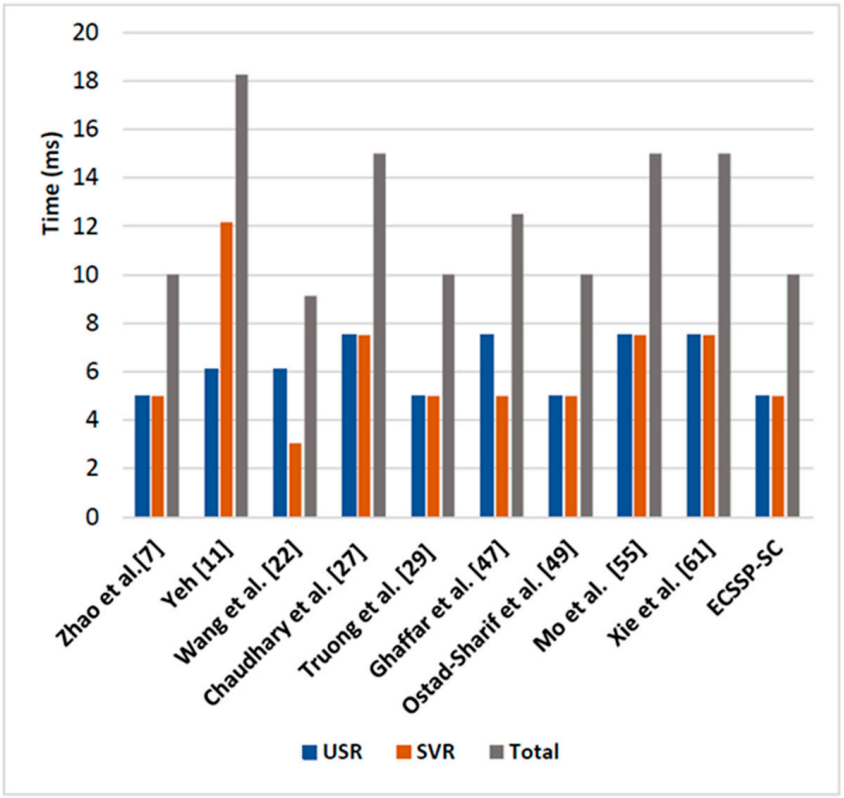

Figure 9. Comparison of the computational time of smart card security protocols.

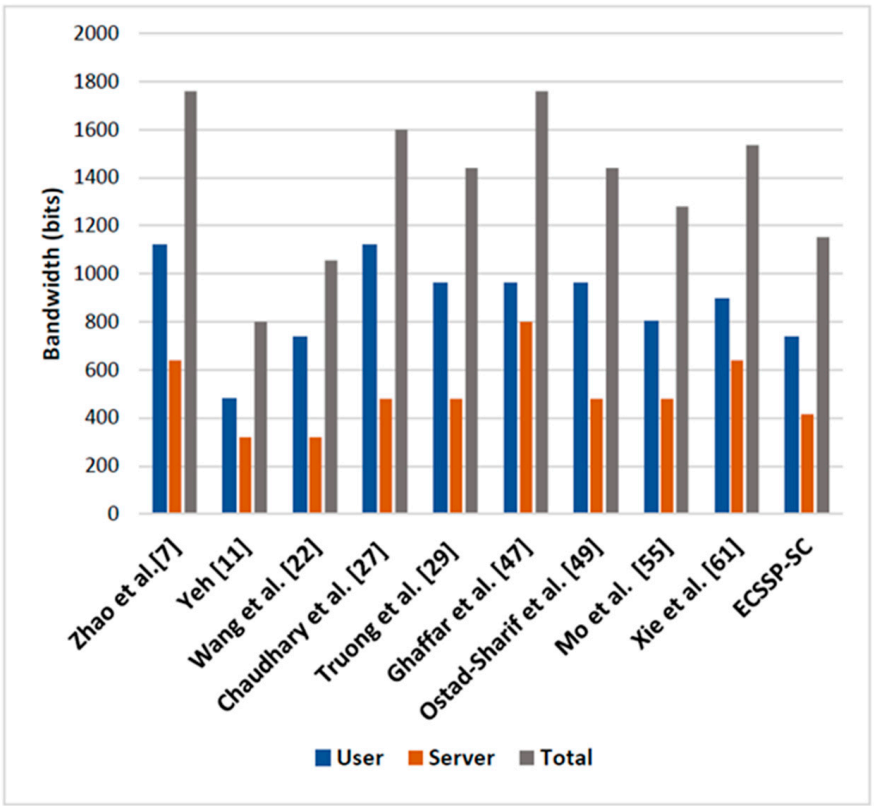

Figure 10. Comparison of the communication cost of smart card security protocols.

Table 5. Security attributes of different smart card security protocols.

Protocol	Security Properties						
	M_{UA}	N_{RP}	F_{WS}	A_{NY}	C_{ON}	I_{NT}	A_{VA}
Zhao et al. [7]	✓	✓	✓	✓	✓	✓	✓
Yeh [11]	×	✓	×	×	×	×	✓
Wang et al. [22]	✓	✓	×	✓	×	×	✓
Chaudhary et al. [27]	✓	✓	✓	✓	✓	✓	✓
Truong et al. [29]	✓	✓	×	✓	×	×	✓
Ghaffar et al. [47]	✓	✓	✓	✓	✓	✓	✓
Ostad-Sharif et al. [49]	✓	✓	×	✓	×	×	✓
Mo et al. [55]	✓	✓	✓	✓	✓	✓	✓
Xie et al. [61]	✓	✓	✓	✓	✓	✓	✓
ECSSP-SC	✓	✓	✓	✓	✓	✓	✓

M_{UA}—mutual authentication, F_{WS}—forward security, N_{RP}—non-repudiation, A_{NY}—anonymity, C_{ON}—confidentiality, I_{NT}—integrity, A_{VA}—availability, ✓—satisfied, ×—not satisfied.

Table 6. Resistance capability of different smart card security protocols against attacks.

Protocol	Resistance to Attacks							
	RP_A	KK_A	US_I	SV_I	IN_S	PG_A	DE_A	MI_A
Zhao et al. [7]	✓	✓	✓	✓	✓	✓	✓	✓
Yeh [11]	✓	✓	×	×	✓	×	×	✓
Wang et al. [22]	✓	✓	✓	✓	✓	×	✓	✓
Chaudhary et al. [27]	✓	✓	✓	✓	✓	✓	✓	✓
Truong et al. [29]	✓	✓	×	×	✓	×	✓	✓
Ghaffar et al. [47]	✓	×	✓	✓	×	✓	✓	✓
Ostad-Sharif et al. [49]	✓	✓	×	×	✓	×	✓	✓
Mo et al. [55]	✓	✓	✓	✓	✓	✓	✓	✓
Xie et al. [61]	✓	✓	✓	✓	✓	✓	✓	✓
ECSSP-SC	✓	✓	✓	✓	✓	✓	✓	✓

RP_A—replay attack, KK_A—known key attack, US_I—user impersonation, SV_I—server impersonation, IN_S—insider attack, PG_A—password guessing attack, DE_A—de-synchronization attack, MI_A—man-in-the-middle attack, ✓—satisfied, ×—not satisfied.

8. Discussion

It can be inferred that the smart card security schemes mentioned in Yeh et al. [11], Wang et al. [22], Truong et al. [29], Ghaffar et al. [47], and Ostad-Sharif et al. [49] either fail to satisfy one or more security attributes listed in Table 5 or fail to resist one or more security attacks mentioned in Table 6. Only the schemes of Zhao et al. [7], Chaudhary et al. [27], Mo et al. [55], and Xie et al. [61], and the proposed ECSSP-SC, successfully satisfy each security attribute listed in Table 5 and can counter every attack enumerated in Table 6. However, it is revealed that the proposed ECSSP-SC consumes the least computational time in comparison to the schemes of Zhao et al. [7], Yeh [11], Wang et al. [22], Chaudhary et al. [27], Truong et al. [29], Mo et al. [55], and Xie et al. [61] Though the scheme of Xie et al. [61] satisfies all the required security features, it consumes three ECPM operations for both the server and the smart card, whereas the ECSSP-SC consumes only two ECPM operations for both the smart card and the server. Thus, ECSSP-SC saves 33.3% computational time compared to the schemes of Chaudhary et al. [27], Mo et al. [55], and Xie at al. [61] and 6% computational time compared to the scheme of Zhao et al. [7] Additionally, the scheme of Xie et al. [61] consumes 896 bits and 640 bits of bandwidth for the smart card and the server, respectively, whereas the proposed ECSSP-SC consumes only 736 bits and 416 bits of bandwidth for the smart card and the server respectively, i.e., saving 17.8% bandwidth for the smart card. Thus, the proposed mutual authentication mechanism based on elliptic curve signcryption is more computational time-efficient than the related schemes and protocols for smart cards. Furthermore, from Figure 10, it is clear that the communication cost of the ECSSP-SC

for both the *USR* and *SVR* is less than that of communication cost consumed in the schemes of Zhao et al. [7], Chaudhary et al. [27], Mo et al. [55], and Xie et al. [61]. The proposed ECSSP-SC consumes 34.5% less bandwidth than the protocol mentioned in Zhao et al. [7].

The smart card security schemes mentioned in the literature survey are based either on modular exponentiation or elliptic curves. The schemes of Yeh et al. [11] and Wang et al. [22] listed in Table 4, Table 5, and Table 6, are based on modular exponentiation, while the other schemes mentioned in [7,27,29,47,49,55,61] are based on elliptic curves, although they do not use signcryption. As mentioned in Table 3, the proposed ECSSP-SC has utilized the signcryption based on an elliptic curve to reduce the computational time and to decrease the communication overhead. No scheme in the literature survey has taken the advantage of combining an elliptic curve with signcryption for smart card security. Results have shown that combining an elliptic curve with signcryption is effective for securing smart card-based systems by satisfying the required security functions, while at the same time consuming less computational and communication cost.

9. Conclusions

A security protocol for smart cards based on the elliptic curve signcryption has been developed and presented in this paper. It has been shown that the presented protocol satisfies mutual authentication, non-repudiation, forward security, anonymity, confidentiality, integrity, and availability attributes. Furthermore, it has the capability to defend against known key attacks, replay attacks, user impersonation, server impersonation, insider attacks, de-synchronization attacks, man-in-the-middle attacks, and password guessing attacks. The analysis of computational time has shown that only two ECPM operations are to be executed by the user. This protocol consumes the least computational cost and communication bandwidth in comparison to the other smart card security protocols. ECSSP-SC reduces the communication overhead by 34.5% and computational time by 6% as compared to other efficient protocols. Since the presented protocol satisfies all the required security functions with the least computational cost and communication bandwidth for the smart card, it outperforms the existing smart card security protocols. The possible future improvements in the research work presented in this paper could be to use hyper elliptic curve cryptography (HECC) for designing efficient signcryption based security protocol for smart cards, because hyper elliptic curve cryptography (HECC) is the upcoming area in cryptography, which delivers an equal level of security to 160 bit ECC using only 80 bits, i.e., the parameter size in HECC is halved compared to ECC.

Author Contributions: Conceptualization, A.K.S., A.S. and A.N.; methodology, A.K.S. and A.S.; software, A.S. and A.K.S.; validation, A.N., A.K.S. and A.S.; formal analysis, A.N., A.K.S. and A.S.; investigation, A.N. and A.S.; resources, A.N. and B.Q.; data curation, A.K.S., A.S. and A.N.; writing—original draft preparation, A.K.S. and A.S.; writing—review and editing, A.N., A.K.S. and B.Q.; visualization, A.S. and A.N.; supervision, A.N. and B.Q.; project administration, A.N. and B.Q.; funding acquisition, B.Q. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank Prince Sultan University, Riyadh, Saudi Arabia for partially funding this research work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mohammed, L.A.; Ramli, A.R.; Prakash, V.; Daud, M.B. Smart card technology: Past, present, and future. *Int. J. Comput. Internet Manag.* **2004**, *12*, 12–22.
2. Fernandes, N.A. Reliable Electronic Certification on Mobile Devices. Master's Thesis, University of Lisbon, Lisbon, Portugal, 2015.
3. Ko, H.; Caytiles, R.D. A Review of Smartcard Security Issues. *J. Secur. Eng.* **2011**, *8*, 359–370.
4. Pippal, R.S.; Jaidhar, C.D.; Tapaswi, S. Security issues in smart card authentication scheme. *Int. J. Comput. Theory Eng.* **2012**, *4*, 206–211. [CrossRef]
5. Singh, A.K.; Patro, B.D.K. Security of Low Computing Power Devices: A Survey of Requirements, Challenges & Possible Solutions. *Cybern. Inf. Technol.* **2019**, *19*, 133–164.

6. Mahanta, H.J.; Azad, A.K.; Khan, A.K. Power analysis attack: A vulnerability to smart card security. In Proceedings of the 2015 International Conference on Signal Processing and Communication Engineering Systems, Guntur, India, 2–3 January 2015; pp. 506–510.
7. Zhao, Y.; Li, S.; Jiang, L. Secure and efficient user authentication scheme based on password and smart card for multiserver environment. *Secur. Commun. Netw.* **2018**, *2018*. [CrossRef]
8. Chang, C.C.; Wu, T.C. Remote password authentication with smart cards. *IEE Proc. E (Comput. Digit. Tech.)* **1991**, *138*, 165–168. [CrossRef]
9. Das, M.L.; Saxena, A.; Gulati, V.P. A dynamic ID-based remote user authentication scheme. *IEEE Trans. Consum. Electron.* **2004**, *50*, 629–631. [CrossRef]
10. Liao, I.E.; Lee, C.C.; Hwang, M.S. Security enhancement for a dynamic ID-based remote user authentication scheme. In Proceedings of the International Conference on Next Generation Web Services Practices (NWeSP'05), Seoul, Korea, 22–26 August 2005; Volume 4.
11. Yeh, K.H.; Su, C.; Lo, N.W.; Li, Y.; Hung, Y.X. Two robust remote user authentication protocols using smart cards. *J. Syst. Softw.* **2010**, *83*, 2556–2565. [CrossRef]
12. Khan, M.K.; Kim, S.K.; Alghathbar, K. Cryptanalysis and security enhancement of a more efficient & secure dynamic ID-based remote user authentication scheme. *Comput. Commun.* **2011**, *34*, 305–309.
13. Liao, Y.P.; Wang, S.S. A secure dynamic ID based remote user authentication scheme for multi-server environment. *Comput. Stand. Interfaces* **2009**, *31*, 24–29. [CrossRef]
14. Hsiang, H.C.; Shih, W.K. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. *Comput. Stand. Interfaces* **2009**, *31*, 1118–1123. [CrossRef]
15. Sood, S.K.; Sarje, A.K.; Singh, K. A secure dynamic identity based authentication protocol for multi-server architecture. *J. Netw. Comput. Appl.* **2011**, *34*, 609–618. [CrossRef]
16. Pippal, R.S.; Jaidhar, C.D.; Tapaswi, S. Robust smart card authentication scheme for multi-server architecture. *Wirel. Pers. Commun.* **2013**, *72*, 729–745. [CrossRef]
17. Yeh, K.H. A provably secure multi-server based authentication scheme. *Wirel. Pers. Commun.* **2014**, *79*, 1621–1634. [CrossRef]
18. Zhang, L.; Tang, S.; Cai, Z. Efficient and flexible password authenticated key agreement for voice over internet protocol session initiation protocol using smart card. *Int. J. Commun. Syst.* **2014**, *27*, 2691–2702. [CrossRef]
19. Farash, M.S.; Attari, M.A. An anonymous and untraceable password-based authentication scheme for session initiation protocol using smart cards. *Int. J. Commun. Syst.* **2016**, *29*, 1956–1967. [CrossRef]
20. Odelu, V.; Das, A.K.; 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*, 2571–2598. [CrossRef]
21. Islam, S.H. Design and analysis of an improved smartcard-based remote user password authentication scheme. *Int. J. Commun. Syst.* **2016**, *29*, 1708–1719. [CrossRef]
22. 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*, 162–178. [CrossRef]
23. Challa, S.; Das, A.K.; Kumari, S.; Odelu, V.; Wu, F.; Li, X. Provably secure three-factor authentication and key agreement scheme for session initiation protocol. *Secur. Commun. Netw.* **2016**, *9*, 5412–5431. [CrossRef]
24. Dhillon, P.K.; Kalra, S. Secure and efficient ECC based SIP authentication scheme for VoIP communications in internet of things. *Multimed. Tools Appl.* **2019**, *78*, 22199–22222. [CrossRef]
25. Reddy, A.G.; Das, A.K.; Yoon, E.J.; Yoo, K.Y. A secure anonymous authentication protocol for mobile services on elliptic curve cryptography. *IEEE Access* **2016**, *4*, 4394–4407. [CrossRef]
26. Wu, H.L.; Chang, C.C.; Chen, L.S. On the Security of a Secure Anonymous Authentication Protocol for Mobile Services on Elliptic Curve Cryptography. In Proceedings of the 6th International Conference on Information Technology: IoT and Smart City, Hong Kong, 29–31 December 2018; pp. 88–91.
27. Chaudhry, S.A.; Naqvi, H.; Mahmood, K.; Ahmad, H.F.; Khan, M.K. An improved remote user authentication scheme using elliptic curve cryptography. *Wirel. Pers. Commun.* **2017**, *96*, 5355–5373. [CrossRef]
28. Huang, B.; Khan, M.K.; Wu, L.; Muhaya, F.T.B.; He, D. An efficient remote user authentication with key agreement scheme using elliptic curve cryptography. *Wirel. Pers. Commun.* **2015**, *85*, 225–240. [CrossRef]
29. Truong, T.T.; Tran, M.T.; Duong, A.D.; Echizen, I. Provable identity based user authentication scheme on ECC in multi-server environment. *Wirel. Pers. Commun.* **2017**, *95*, 2785–2801. [CrossRef]

30. Chandrakar, P.; Om, H. An efficient two-factor remote user authentication and session key agreement scheme using rabin cryptosystem. *Arab. J. Sci. Eng.* **2018**, *43*, 661–673. [CrossRef]
31. Jiang, Q.; Zeadally, S.; Ma, J.; He, D. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access* **2017**, *5*, 3376–3392. [CrossRef]
32. Qiu, S.; Xu, G.; Ahmad, H.; Wang, L. A robust mutual authentication scheme based on elliptic curve cryptography for telecare medical information systems. *IEEE Access* **2017**, *6*, 7452–7463. [CrossRef]
33. Zhang, Y.; Xie, K.; Ruan, O. An improved and efficient mutual authentication scheme for session initiation protocol. *PLoS ONE* **2019**, *14*, e0213688. [CrossRef]
34. Kumari, S.; Karuppiah, M.; Das, A.K.; Li, X.; Wu, F.; Gupta, V. Design of a secure anonymity-preserving authentication scheme for session initiation protocol using elliptic curve cryptography. *J. Ambient Intell. Humaniz. Comput.* **2018**, *9*, 643–653. [CrossRef]
35. Qiu, S.; Xu, G.; Ahmad, H.; Xu, G.; Qiu, X.; Xu, H. An Improved Lightweight Two-Factor Authentication and Key Agreement Protocol with Dynamic Identity Based on Elliptic Curve Cryptography. *TIIS* **2019**, *13*, 978–1002.
36. Limbasiya, T.; Soni, M.; Mishra, S.K. Advanced formal authentication protocol using smart cards for network applicants. *Comput. Electr. Eng.* **2018**, *66*, 50–63. [CrossRef]
37. Dharminder, D.; Rana, S.; Kundu, N.; Mishra, D. Construction of lightweight authentication scheme for network applicants using smart cards. *Sādhanā* **2020**, *45*, 15. [CrossRef]
38. Sureshkumar, V.; Amin, R.; Anitha, R. A robust mutual authentication scheme for session initiation protocol with key establishment. *Peer Netw. Appl.* **2020**, *11*, 900–916. [CrossRef]
39. Sourav, S.; Odelu, V.; Prasath, R. Enhanced session initiation protocols for emergency healthcare applications. In *International Symposium on Security in Computing and Communication*; Springer: Singapore, 2018; pp. 278–289.
40. Qiu, S.; Xu, G.; Ahmad, H.; Guo, Y. An enhanced password authentication scheme for session initiation protocol with perfect forward secrecy. *PLoS ONE* **2018**, *13*, e0194072. [CrossRef]
41. Nikooghadam, M.; Amintoosi, H. A secure and robust elliptic curve cryptography-based mutual authentication scheme for session initiation protocol. *Secur. Priv.* **2020**, *3*, e92. [CrossRef]
42. Shouqi, C.; Wanrong, L.; Liling, C.; Xin, H.; Zhiyong, J. An Improved Authentication Protocol Using Smart Cards for the Internet of Things. *IEEE Access* **2019**, *7*, 157284–157292. [CrossRef]
43. Zhao, Y.; Li, S.; Jiang, L.; Liu, T. Security-enhanced three-factor remote user authentication scheme based on Chebyshev chaotic maps. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1–12. [CrossRef]
44. Dharminder, D.; Gupta, P. Security analysis and application of Chebyshev Chaotic map in the authentication protocols. *Int. J. Comput. Appl.* **2019**, 1–9. [CrossRef]
45. Zheng, L.; Xue, Y.; Zhang, L.; Zhang, R. Mutual Authentication Protocol for RFID based on ECC. In Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 21–24 July 2017; Volume 2, pp. 320–323.
46. Wang, F.; Xu, G.; Xu, G.; Wang, Y.; Peng, J. A Robust IoT-Based Three-Factor Authentication Scheme for Cloud Computing Resistant to Session Key Exposure. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 3805058. [CrossRef]
47. Ghaffar, Z.; Ahmed, S.; Mahmood, K.; Islam, S.H.; Hassan, M.M.; Fortino, G. An Improved Authentication Scheme for Remote Data Access and Sharing over Cloud Storage in Cyber-Physical-Social-Systems. *IEEE Access* **2020**, *8*, 47144–47160. [CrossRef]
48. Yu, Y.; Hu, L.; Chu, J. A Secure Authentication and Key Agreement Scheme for IoT-Based Cloud Computing Environment. *Symmetry* **2020**, *12*, 150. [CrossRef]
49. Ostad-Sharif, A.; Abbasinezhad-Mood, D.; Nikooghadam, M. A robust and efficient ECC-based mutual authentication and session key generation scheme for healthcare applications. *J. Med. Syst.* **2019**, *43*, 10. [CrossRef] [PubMed]
50. Kumari, S.; Chaudhary, P.; Chen, C.M.; Khan, M.K. Questioning key compromise attack on Ostad-Sharif et al.'s authentication and session key generation scheme for healthcare applications. *IEEE Access* **2019**, *7*, 39717–39720. [CrossRef]
51. Choudhary, K.; Gaba, G.S.; Butun, I.; Kumar, P. MAKE-IT—A Lightweight Mutual Authentication and Key Exchange Protocol for Industrial Internet of Things. *Sensors* **2020**, *20*, 5166. [CrossRef]

52. Mandal, S.; Bera, B.; Sutrala, A.K.; Das, A.K.; Choo, K.K.R.; Park, Y. Certificateless-Signcryption-Based three-factor user access control scheme for IoT environment. *IEEE Internet Things J.* **2020**, *7*, 3184–3197. [CrossRef]
53. Rajasekar, V.; Premalatha, J.; Sathya, K. Multi-factor signcryption scheme for secure authentication using hyper elliptic curve cryptography and bio-hash function. *Bull. Pol. Acad. Sci. Tech. Sci.* **2020**, *68*, 923–935.
54. Martínez, V.G.; Encinas, L.H. Developing ECC applications in Java Card. In Proceedings of the 2013 9th International Conference on Information Assurance and Security (IAS), Gammarth, Tunisia, 4–6 December 2013; pp. 114–120.
55. Mo, J.; Hu, Z.; Chen, H.; Shen, W. An efficient and provably secure anonymous user authentication and key agreement for mobile cloud computing. *Wirel. Commun. Mob. Comput.* **2019**, *2019*. [CrossRef]
56. Lauter, K.E.; Stange, K.E. The elliptic curve discrete logarithm problem and equivalent hard problems for elliptic divisibility sequences. In *International Workshop on Selected Areas in Cryptography*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 309–327.
57. Shparlinski, I. Computational Diffie-Hellman problem. In *Encyclopedia of Cryptography and Security*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 240–244.
58. Boneh, D. The decision diffie-hellman problem. In *International Algorithmic Number Theory Symposium*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 48–63.
59. Zheng, Y.; Imai, H. How to construct efficient signcryption schemes on elliptic curves. *Inf. Process. Lett.* **1998**, *68*, 227–233. [CrossRef]
60. Ouafi, K.; Phan, R.C.W. Traceable privacy of recent provably-secure RFID protocols. In *International Conference on Applied Cryptography and Network Security*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 479–489.
61. Xie, Q.; Wong, D.S.; 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. Forensics Secur.* **2017**, *12*, 1382–1392. [CrossRef]