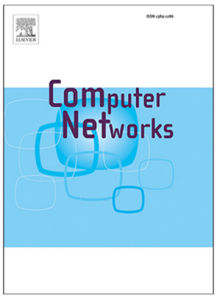

ECCPWS: An ECC-based protocol for WBAN systems

Fatemeh Pirmoradian, Masoumeh Safkhani, Seyed Mohammad Dakhilalian

PII: S1389-1286(23)00043-9
DOI: https://doi.org/10.1016/j.comnet.2023.109598
Reference: COMPNW 109598

To appear in: *Computer Networks*

Received date : 3 October 2021
Revised date : 15 December 2022
Accepted date : 26 January 2023

Please cite this article as: F. Pirmoradian, M. Safkhani and S.M. Dakhilalian, ECCPWS: An ECC-based protocol for WBAN systems, *Computer Networks* (2023), doi: https://doi.org/10.1016/j.comnet.2023.109598.

ECCPWS: An ECC-based Protocol for WBAN Systems

Fatemeh Pirmoradian[a], Masoumeh Safkhani[b,c] and Seyed Mohammad Dakhilalian[a,*]

[a]Department of Electrical and Computer Engineering, Isfahan University of Technology (IUT), Isfahan, Iran, Postal code: 84156-83111

[b]Department of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran, Postal code: 16788-15811, Tel/fax: +98-21-22970117

[c]School of Computer Science, Institute for Research in Fundamental Sciences (IPM), P. O. Box 19395-5746, Tehran, Iran

ARTICLE INFO

Keywords:
Internet of Things
Wearable Health Monitoring Systems
Elliptic Curve Cryptography
Passive Insider Secret Disclosure Attack
Replay Attack
Impersonation Attack
Real-Or-Random (ROR) Model
BAN Logic
AVISPA Tool
Scyther Tool

ABSTRACT

The development of Wireless Body Area Network (WBAN) and Wearable Health Monitoring Systems (WHMS) play a key role in healthcare monitoring. WBAN includes medical sensors that monitor vital signs of patients, collect data and usually transmit them to medical servers via wireless channels. Therefore, the patient's sensitive information sent over the channel can be vulnerable to various attacks. Hence, designing lightweight authentication security protocols for these systems with the lowest computational and communication costs has become a major challenge. Recently, Sowjanya *et al.* presented a lightweight authentication scheme for WHMS based on Elliptic Curve Cryptography (ECC), which provides optimal security features and low storage, communication and computational costs. In this paper, the security of their scheme is evaluated and passive insider secret disclosure and replay attacks against their scheme are presented. It is obvious that other attacks such as desynchronization attack and impersonation attack can be applied to this protocol by obtaining the secret value of network manager. The complexity of our proposed attacks is just one run of the protocol and their success probability equals to one. Finally, by remedying the Sowjanya *et al.*'s protocol, a lightweight ECC-based authentication scheme called ECCPWS is proposed. Moreover, the proposed protocol's security proof is performed via informal methods and also formally through Real-Or-Random (ROR) model, BAN logic, Scyther and AVISPA tools. The security verification results of ECCPWS show that the ECCPWS has complete security against various security vulnerabilities and attacks.

1. Introduction

The Internet of Things (IoT) network is useful for connecting objects to each other through the Internet [1, 2, 3]. This network has deployed in recent years, therefore a large number of sensors have been used to collect data in various fields of IoT [4, 5]. One of these fields is medical services on the Internet platform, which reduce the cost of going to the hospital and save time [6]. One of the essential technologies in IoT is Telecare Medicine Information System (TMIS) [7]. Great progress in the field of medical services have established Wearable Health Monitoring Systems (WHMS) [8]. As shown in Fig.1, a WHMS includes three participants namely; Network Manager (NM), Application Server (AS) and WBAN User (U). Wireless Body Area Network (WBAN) plays a fundamental role in medical industry to getting informed of the health status of patients in WHMS [9, 10]. WBAN includes the various medical sensors, that are equipped on the body of patients, sense patient's vital signs parameters such as body temperature, blood pressure and etc. and transmit measured data to medical servers through Internet [11, 12]. Since, communication sensors are used to measure patient's sensitive information on insecure channels, therefore privacy of these information, user's anonymity and other security features have become a great challenge [13, 14, 15]. Since, this medical information is vulnerable to various security attacks [16], therefore, various schemes have been developed to guarantee the security of data [17, 18, 19]. In this line, recently, Sowjanya *et al.* [20] proposed a lightweight authentication scheme for WBAN based on ECC.

1.1. Main Contribution

Lately, Sowjanya *et al.* [20] presented a lightweight ECC-based anonymous authentication protocol for WHMS. In this paper, we show that this anonymous authentication scheme is vulnerable to passive insider secret disclosure and replay attacks. Then, we remedy the weaknesses of this scheme, that lead to propose a new lightweight authentication scheme based on ECC named ECCPWS. Therefore, main contributions of this paper are summarized as follows:

- The vulnerabilities of Sowjanya *et al.*'s anonymous authentication scheme against the passive insider secret disclosure and replay attacks are presented.

Mdalian@iut.ac.ir (Seyed Mohammad Dakhilalian[a,*])
ORCID(s):

- A lightweight ECC-based authentication scheme is suggested for WHMS, named ECCPWS, which resolves all security vulnerabilities of its predecessor scheme.
- The security proof of ECCPWS is accomplished with informal methods and also formal methods such as Real-Or-Random (ROR) model and Burrows-Abadi-Needham (BAN) logic.
- The security verification of ECCPWS is performed through Scyther and AVISPA tools.

1.2. Organization

The rest of this paper is structured as follows: In sec.2, related work in this field are presented. The preliminaries, required contexts and Sowjanya *et al.*'s scheme are explained in sec.3 and sec.4, respectively. Also, the security analysis of Sowjanya *et al.*'s protocol is presented in sec.5 and then, a lightweight ECC-based authentication protocol is proposed for WHMS in sec.6, which this scheme eliminates the vulnerabilities of the Sowjanya *et al.*'s scheme. The security analysis of proposed protocol informally and formally via ROR model, BAN logic, Scyther tool and AVISPA tool are discussed in sec.7. Sec.8 compares performance of the improved protocol i.e. ECCPWS with recent related protocols and ultimately, sec.9 concludes the paper with concluding remarks.

2. Related work

Lamport was the first person that proposed a password-based authentication scheme in insecure communication channel for WBAN networks in 1981 [21]. Also, Azim *et al.* presented a key agreement protocol based on ECC for wireless LAN [22]. Then, Otto *et al.* introduced an architecture for wireless body area network for health monitoring in 2006 [23]. Subsequently, Boyle *et al.* suggested various security protocols for employing in wireless sensor networks [24]. Later, Preneel *et al.* presented a paper about security concepts required for wireless networks in 2008 [25]. Then, Le *et al.* proposed a security scheme based on public key cryptography for wireless sensor networks in healthcare [26]. Subsequently, Liu *et al.* proposed connective security workmanship for wireless body networks in 2010 [27]. Also, in 2011, Mana *et al.* presented another scheme for WBAN [28]. Then, Yeh *et al.* presented a protocol based on modular exponential operations for wireless health monitoring systems [29]. In 2014, Zhao presented a protocol for wireless body networks based on ECC cryptosystems which has many security loopholes such as desynchronization problems [30]. Consequently, He and Zeadly presented an authentication scheme for best living system [31]. Thereafter, Yessad *et al.* introduced a trustworthy patient body motion based on verification approach for medical body area networks in 2017 [9] and Wu *et al.* proposed a lightweight and privacy preserving mutual authentication scheme for wearable devices assisted by the cloud server [32]. Independently, Amin *et al.* proposed a huge medical sensor network in 2018 [7], that its security is evaluated using BAN logic and AVISPA tool. Following that, Jiang *et al.* demonstrated that the Amin's scheme [7] is vulnerable to a variety of attacks. They also proposed a verification scheme to improve their protocol security loopholes for wearable health monitoring systems. In recent years, many schemes in the field of authentication of WBAN have been proposed. For example, Liu *et al.* introduced a 1-round verification scheme for wireless body networks in 2016 [33]. In 2017, Li *et al.* showed that their scheme has vulnerabilities against several attacks such as Denial of Service (DoS) attack, stolen verifier attack and so on [34]. They also proposed another 1-round verification scheme for wireless body networks. Although Li *et al.* claimed that their protocol provides complete security, but unfortunately Sowjanya *et al.* showed that this scheme has various security weaknesses such as lack of key control, perfect forward secrecy and mutual authentication among network manager (NM) and WBAN client (U). They also proposed an ECC-based enhanced anonymous authentication protocol for WHMS in 2019 [20]. In this line and in this paper, we show that the Sowjanya *et al.*'s scheme has vulnerabilities against passive insider secret disclosure and replay attacks. Then, we propose an ECC-based scheme called ECCPWS, that improves the security weaknesses related to the Sowjanya *et al.*'s protocol. We also prove that the ECCPWS resists against known active and passive insider and outsider attackers.

3. Preliminaries

The current section explains the used notations, the major concepts in ECC and framework of WHMS.

3.1. Notations

All the used notations in this paper are represented in Table 1.

3.2. Elliptic curve based cryptography

In summary, this part reviews the basic concepts required in this paper for Elliptic Curve Cryptography (ECC). ECC is a kind of public key algorithms, which provides security similar to Discrete Logarithm systems and public key algorithms like RSA but with shorter keys (160 - 256 bits vs 1024 - 3072 bits). Elliptic curve has advantages such as better bandwidth, less computational complexity, shorter signatures and keys with fewer bits. The elliptic curve is a set of all tuples $(x, y) \in Z_p$, which are applied to the equation $y^2 = x^3 + ax + b \ mod \ p$, where

$a, b \in \mathbb{Z}_p$ and O as infinity imaginary point with the property $4a^3 + 27b^2 \neq 0 \ mod \ p$. This equation is called the Weierstrass equation [35, 36, 37].

3.2.1. Group operations on elliptic curve

One feature of ECC is that it takes two points on elliptic curve and "adds" these two points to achieve the third point. As a result, there are two cases of this type of addition, that are distinguished from each other as below [37].

1. Construction the third point by adding two distinct points, which is called the point addition.
2. Construction the third point by adding a point to itself, which is called point redoubling.

- **Point Addition** or $P + Q$: Suppose $E : Y^2 = X^3 + AX + B$ is an elliptic curve and the points P and Q are two separate points on elliptic curve. If $P = O$ or $Q = O$, the equations $P + Q = Q$ or $P + Q = P$ are established, respectively. Also, if the two points P and Q are defined as $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ and $x_1 = x_2$ and $y_1 = -y_2$, then the sum of two points on elliptic curve is defined as $P + Q = O$. Otherwise, the parameter λ is defined as follows, assuming $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$ and as a result, the equation $P + Q = (x_3, y_3)$ is established.

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & if \ P \neq Q \\ \frac{3x_1^2 + A}{2y_1} & if \ P = Q \end{cases}$$

- **Point Redoubling** or $P + P$: In this case, suppose E is an elliptic curve with point O at infinity, P and Q are two separate points on elliptic curve and line $l_{P,Q}$ passing through points P and Q. If the point Q is closer to the point P, the line $l_{P,Q}$ is tangent to E at point P. So, if we want to add the point P with itself, consider the line $l_{P,Q}$ that is tangent to E at the point P. Hence, $P + Q$, where $P = Q$, can be written as $R = P + P = 2.P$.

- **Scalar Point Multiplication** or $n.P$: Suppose E is an elliptic curve, P is a point on the elliptic curve E and n is an integer. The multiplication of the scalar points i.e. $n.P$ is calculated as the sum of the points P with itself n times ($n.P = P + P + ... + P$) and as the iterative sum.

3.3. Mathematical assumptions based on ECC

The security of ECC-based security protocols is related to the hard problems on ECC, which some of these assumptions are described as below:

- The Elliptic Curve Discrete Logarithm Problem (ECDLP): Let E is an elliptic curve in the finite field F_p and two points P and Q are the points in group $E(F_p)$. Then, the integer n exists, so that the equation $Q = n.P$ is established. Therefore, solving the problem of discrete logarithm in elliptic curve is the problem of determining n with respect to the equation $Q = P + P + P + ... + P = n.P$. As a result, the value of secret n is equal to the discrete logarithm of elliptic curve of the point Q with respect to the point P as $n = log_P(Q)$. In fact, the smallest positive integer n with this property is called the discrete logarithm Q at base P [37, 38].

- The Elliptic Curve Computational Diffie-Hellman Problem (ECCDHP): ECCDHP refers to the exchange of the diffie-hellman key on the elliptic curve. Suppose Alice and Bob are looking for a common key. As a result, they select $E(F_p)$ and $P \in (F_p)$ as an elliptic curve and a point P in the finite field associated with the elliptic curve, respectively. Of course, this curve must be chosen in such that, it is difficult to solve the problem of discrete logarithm. Alice and Bob choose the secret key n_1 and n_2, respectively. In this case, $n_1 n_2.P$ will be the common key of Alice and Bob. Due to the difficulty of solving the discrete logarithm problem in this curve, given P, $n_1.P$ and $n_2.P$, retreiveing of $n_1 n_2.P$ is difficult. The problem of finding $n_1 n_2.P$ using P, $n_1.P$, and $n_2.P$ in an elliptic curve is called the Elliptic Curve Computational Diffie Hellman Problem or ECCDHP.

- The Collision-resistant One-way Hash Function: Let $H : \{0,1\}^* \rightarrow \{0,1\}^l$ is a deterministic mathematical function with input of arbitrary length $x \in \{0,1\}^*$ and output with the fixed length (l-bits) i.e. $h(x) \in \{0,1\}^l$. If $Adv_A^{Hash}(t)$ is the advantage of an adversary $\mathcal{A}$ in finding a collision in the hash function, the value of $Adv_A^{Hash}(t)$ is $Adv_A^{Hash}(t) = Pr[(a,b) \in_R \mathcal{A} : a \neq b, h(a) = h(b)]$, where $Pr[X]$ and $(a,b) \in_R \mathcal{A}$, are assigned to the probability of an event X and the pair (a,b) is selected by $\mathcal{A}$ randomly, respectively [39, 40]. Therefore, the probability of winning $\mathcal{A}$ at runtime t is calculated as $Adv_A^{Hash}(t)$, which must be true for $Adv_A^{Hash}(t) \leq \varepsilon$ with insignificant amount of ε [39, 40].

- The Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP): Suppose $P \in E_p(a,b)$ is a point on an elliptic curve $E_p(a,b)$ in the finite field. If there is the equation $k_3 = k_1.k_2$ where $(P, k_1.P, k_2.P, k_3.P)$, $Z_p^* = \{1, 2, ..., p-1\}$ and

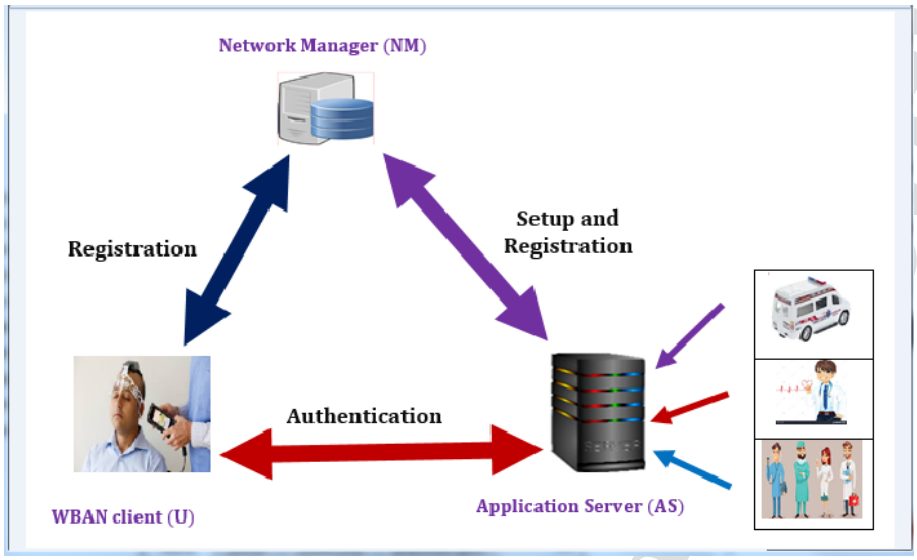

Figure 1: A generic architecture for Wearable Health Monitoring Systems (WHMS).

Table 1
The used notations in this paper.

Notation	Description
NM	Network Manager
AS	Application Server
U	User
p	Prime number in the finite field
E	Elliptic curve with order of p
G_E	Additive group with order of q
B	Generator of G_E
$h()$	Hash function $h : \{0,1\}\times G_E \rightarrow Z_q^*$
Z_q	Integers $\{0, 1, 2, ..., q-1\}$ in the finite field
Z_q^*	Z_q - $\{0\}$
ID_U	Identity of user U_i
SK	The session key
ΔT	The maximum communication transmission delay
K_{UA}	The shared secret key between the U and server AS
(S_{AS}, PK_{AS})	Private/public key pair of Application Server
(S_{NM}, PK_{NM})	Private/public key pair of Network Manager
(S_U, PK_U)	Private/public key pair of User
$(\perp)$	The null value
$\mathcal{O}_U^t$	Oracle of participant U at time t
$\mathcal{O}_{NM}^u$	Oracle of participant NM at time u
$\mathcal{O}_{AS}^v$	Oracle of participant AS at time v
$Game_i$	The game number i used in ROR model
T_1, T_2, T_3, T_4	The current timestamps
sid	The session identification
q_{hash}	The number of hash queries
$\|Hash\|$	Range space of hash function $H(.)$
T_h	Execution time of hash function
T_s	Execution time of symmetric encryption and decryption
T_m	Execution time of scalar point multiplication based on ECC
$\mathcal{A}$	Adversary
$(.)$	Scalar point multiplication based on ECC

$k_1, k_2, k_3 \in Z_p^*$, then the Decisional Diffie-Hellman Problem in Elliptic Curve or ECDDHP is established. If the value of p as a prime number is selected at least 160 bits, solving ECDDHP is computationally impossible.

3.4. WHMS System Model

WHMS consists of several devices and medical sensors, which communicate to each other via wireless channels. As shown in Fig. 1, there are three participants such as Network Manager (NM), Application Server (AS) and Users (U) in this model. The NM is responsible for setup of system, the registration process and also consists of specialists, medical clinics, hospitals and health centers [41, 42]. In this model, medical servers are considered as separate entities with private keys, that protect the privacy of users [43]. Each user must be registered in the NM through a secure channel, when connected to the system. Also, the server NM will generate necessary information of each user and transmits them back through a secure channel [6]. Therefore, users can use their private key and necessary information to log into the AS as remotely to be authenticated mutually and produce session key for secure communications in public and insecure channels. Each entity of WHMS is described briefly as below [18, 20].

- User (U): The final goal of user is access to the medical services provided by the AS. At first, the user must be registered in the server NM. After the successful registration, the user must be authenticated using the AS in order to use these services.
- Application Server (AS): The AS must be registered and authenticated by the server NM until offers medical services to the legal users. The AS can include hospitals, specialist doctors and clinics.
- Network manager (NM): The task of this server is registration of users to access to the medical services, registration of the AS and giving permission to users to use the medical services which provided by AS.

4. Sowjanya *et al.*'s protocol

Recently, Sowjanya *et al.* [20] presented an improved version of Li *et al.*'s [34] authentication scheme for wearable health monitoring systems to fix the security loopholes of their scheme. Analogous, the Sowjanya *et al.*'s protocol consists of three participants namely; User (U), Application Server (AS) and Network Manager (NM). Sowjanya *et al.*'s scheme [20] consists of three phases called Initialization, Registration and Authentication phases, which the last two are shown in Fig. 2 and Fig. 3, respectively.

4.1. Initialization phase

In this phase of protocol, the server NM generates the protocol parameters, which are defined in Table 1 as follows [20]:

- **Step 1.** The server NM selects a hash function such as $h : \{0,1\} \times G_E \rightarrow {Z_q}^*$.
- **Step 2.** The server NM selects its private/public keys (S_{NM}, PK_{NM}), so that $PK_{NM} = S_{NM}.B$ and $S_{NM} \in {Z_q}^*$.
- **Step 3.** Subsequently, the protocol public parameters such as $\{G_E, p, q, B, h, PK_{NM}\}$ are generated by the server NM.
- **Step 4.** The user U selects its private/public keys as (S_U, PK_U), so that $PK_U = S_U.B$.
- **Step 5.** The server AS selects its private/public keys as (S_{AS}, PK_{AS}), so that $PK_{AS} = S_{AS}.B$.
- **Step 6.** The user U multiplies its ID_U by the generator B and obtains identity parameter $\{ID_U.B\}$ as an element of the elliptic curve group.

4.2. Registration phase

In order to have a legitimate user in WHMS, every user must be registered in the server NM to be able to access the medical services. The registration phase of Sowjanya *et al.*'s protocol is depicted with the following steps in Fig. 2:

- **Step 1.** The user U chooses its ID_U, the public key PK_U and sends registration request $\{ID_U.B, PK_U\}$ to the server NM through a secure and private channel.
- **Step 2.** After receiving the request message, the NM checks out the validity of message $\{ID_U.B\}$. Next, the NM defines parameter "$right$" and picks out a random number $k \in {Z_q}^*$. Afterwards, it computes $Ind_U = k.B$, $U_U = kPK_U$, $x = h(U_U, right)$, $v = xS_{NM}$ and then transmits $\{Ind_U, right, v\}$ to the user through a secure channel.
- **Step 3.** After the message $\{Ind_U, right, v\}$ was received, the user computes $U_U = S_U Ind_U$ and $x = h(U_U, right)$. Also, the U checks out whether the equation $v.B \stackrel{?}{=} xPK_{NM}$ is or not. If the user does not find any match, then rejects the registration answer and stops the protocol. Otherwise, the U stores message $\{Ind_U, right, v\}$ in its own database safely and then calculates $K_{UA} = S_U PK_{AS}$, where K_{UA} is the shared secret key between U and AS. It is obvious that the AS computes this shared secret key as $K_{UA} = S_{AS} PK_U$, which equals to computed key at the user side. This is why $K_{UA} = S_U PK_{AS} = S_U S_{AS}.B = S_{AS} S_U.B = S_{AS} PK_U$.

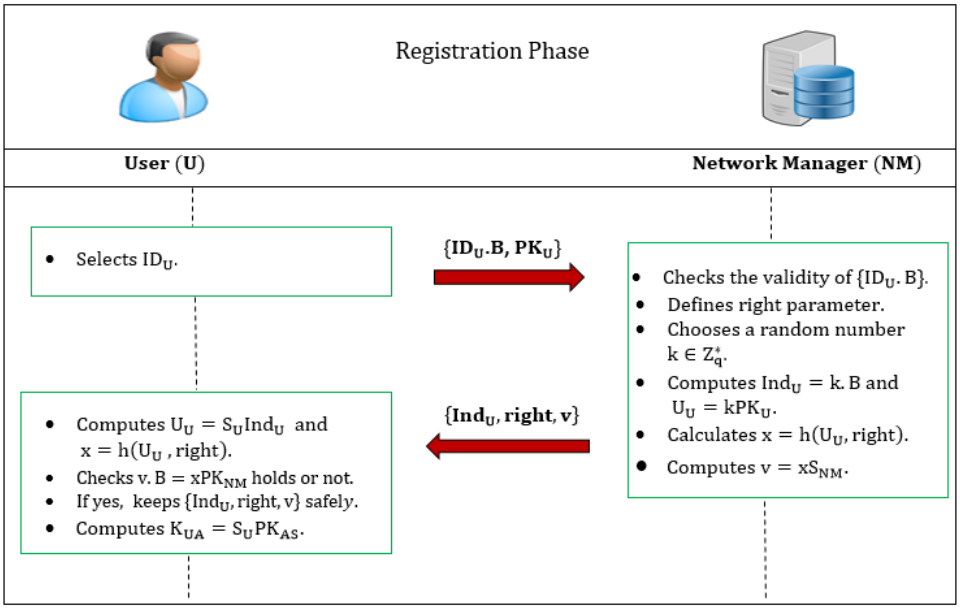

Figure 2: The Registration phase of Sowjanya *et al.*'s protocol [20].

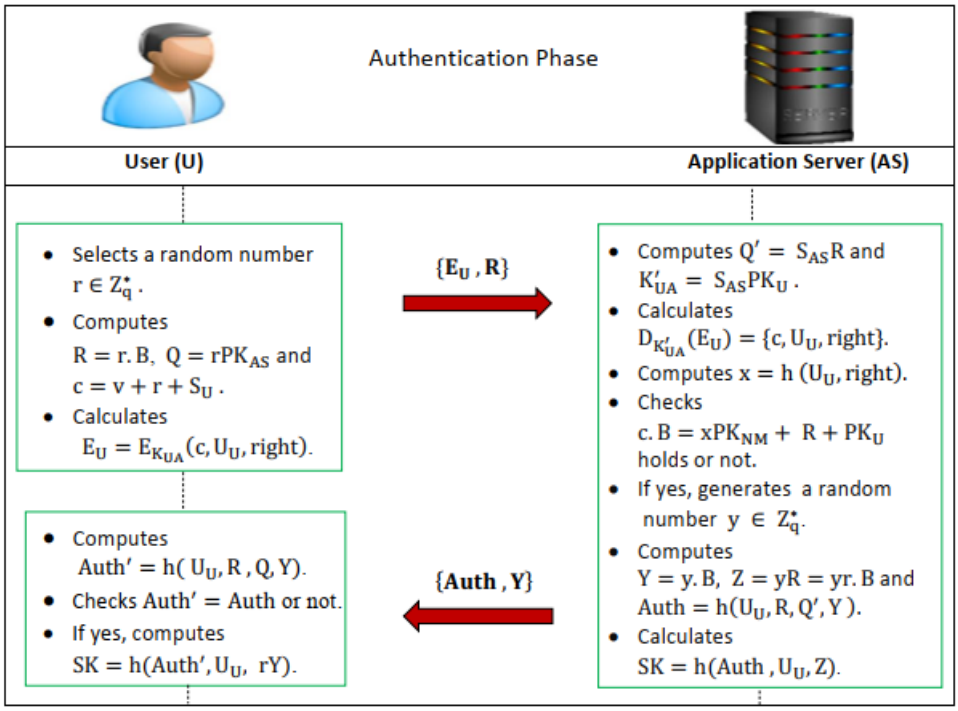

Figure 3: The Authentication phase of Sowjanya *et al.*'s protocol [20].

4.3. Authentication phase

In order to retrieve medical services, the server AS and user U must be authenticated to each other. The authentication phase starts by the user U as follows (see Fig. 3):

- **Step 1.** The user U selects a random number $r \in Z_q^*$ and computes $R = r.B$, $Q = rPK_{AS}$, $E_U = E_{K_{UA}}(c, U_U, right)$ and $c = v + r + S_U$. Next, it sends the authentication request $\{E_U, R\}$ to the AS.
- **Step 2.** After the message $\{E_U, R\}$ was received, the AS calculates $Q' = S_{AS}R$, $K'_{UA} = S_{AS}PK_U$ and $D_{K'_{UA}}(E_U) = \{c, U_U, right\}$ and obtains $\{c, U_U, right\}$.
- **Step 3.** The server AS computes $x = h(U_U, right)$ and verifies whether $c.B \stackrel{?}{=} xPK_{NM} + R + PK_U$. If this equation does not hold, the session ends. Otherwise, the server AS selects a random number $y \in Z_q^*$ and computes $Y = y.B$, $Z = yR = yr.B$, $Auth = h(U_U, R, Q', Y)$ and then $SK = h(Auth, U_U, Z)$ as the session key. At last, the server AS sends the message $\{Auth, Y\}$ to the user U in response to the authentication request.
- **Step 4.** Once received the message $\{Auth, Y\}$, the user computes $Auth' = h(U_U, R, Q, Y)$. Then it verifies whether $Auth' \stackrel{?}{=} Auth$. If it is not, the session terminates. Otherwise, the user U validates the server AS as a valid server and computes $SK = h(Auth', U_U, rY)$ as the session key. Since $rY = ry.B = yr.B = Z$, it is obvious the generated session key in the U's side is equal to generated session key in the AS's side.

Algorithm 1 The algorithm of passive insider secret disclosure attack to retrieve the secret value of NM in the Sowjanya *et al.*'s authentication scheme.

The insider adversary which is a legitimate user, starts one registration session of the protocol by sending its $\{ID_U.B, PK_U\}$ and receiving NM's response i.e. $\{Ind_U, right, v\}$. Then the adversary does as follows:

1. Computes $U_U = S_U Ind_U$ using its S_U and the received Ind_U from the registration channel;
2. Calculates the value of $x = h(U_U, right)$ using its computed U_U in the previous step and the received $right$ from the registration channel;
3. Computes the inverse of x, namely x^{-1};
4. The adversary obtains the value of S_{NM} (i.e. the NM's secret value) by using the received v from the registration channel and computed x^{-1} in previous step as vx^{-1};
5. Returns S_{NM};

5. Security analysis of Sowjanya *et al.*'s protocol

Here, we show that the Sowjanya *et al.*'s protocol is vulnerable to passive insider secret disclosure and replay attacks.

5.1. Passive insider secret disclosure attack

In this type of the attack, the attacker which is a legitimate user of the system, acts passively and only gets the exchanged messages among participants. Then, s/he performs offline computations on the exchanged messages to retrieve the secret values.

5.1.1. Adversary model

In the kind of passive attacks, the adversary carries out its intended attack by only receiving the exchanged messages and doing offline computations on the messages [44, 45, 46]. It worth noting that an insider attacker has access to the secure channel and can observe the exchanged messages in the secure channels such as the channel used in the registration phase of Sowjanya *et al.*'s protocol. Disclosing the secret values enables the attacker to perform any other attacks. Passive insider secret disclosure attack as shown in Algorithm 1 has two phases as below:

1. **Learning phase:** In this phase, the insider attacker (i.e., the attacker which is a legal user and has registered in the system and has S_U) starts the registration session of the protocol and sends its $\{ID_U.B, PK_U\}$ and obtains the NM's response i.e.$\{Ind_U, right, v\}$.
2. **Secret Disclosure phase:** In this phase, the insider attacker finds the secret value of network manager i.e. (S_{NM}) as follows:
   - Computes $U_U = S_U Ind_U$ using its S_U and the received Ind_U from the channel;
   - Calculates the value of $x = h(U_U, right)$ using computed U_U in previous step and the re-

ceived *right* from the registration channel;

- Computes the inverse of x, namely x^{-1};
- Computes the value of S_{NM} using the received v from the registration channel and computed x^{-1} in previous step as vx^{-1};

It is obvious the adversary can carry out other attacks such as desynchronization and impersonation attacks against this protocol by obtaining the secret value of network manager. The complexity of passive insider secret disclosure attack is just one run of the protocol and its success probability is equal to one.

5.2. Replay attack

In this type of attack [47, 48], if the adversary $\mathcal{A}$ reuses the authentication request $\{E_U, R\}$ from the previous session, the server AS will not understand that the messages are not fresh and the adversary $\mathcal{A}$ will be authenticated to the AS due to the lack of timestamps in the calulation of exchanged messages. However, finally they can not calculate the related session key. Because, the random values y and r which generated by AS and U are unique for each session. As a result, the attacker cannot access to these random values generated by the user and server for each session. Therefore, the Sowjanya *et al.*'s protocol is vulnerable to replay attacks.

6. ECCPWS: Proposed protocol

To eliminate the weaknesses of the Sowjanya *et al.*'s scheme, we propose an improved version of it for WHMS, called ECCPWS. Similar to the Sowjanya *et al.*'s protocol, ECCPWS contains three kind of participants NM, U and AS and also consists of three phases that are Initialization, Registration and Authentication phases, that the last two are shown in Fig. 4 and Fig. 5, respectively.

6.1. Initialization phase

In this phase of ECCPWS, the server NM publishes protocol parameters, which are represented in Table 1 by performing the following steps:

- **Step 1.** The server NM selects a hash function such as $h : \{0,1\} \times G_E \rightarrow {Z_q}^*$.
- **Step 2.** The server NM selects its private/public keys as (S_{NM}, PK_{NM}) where $PK_{NM} = S_{NM}.B$ and $S_{NM} \in {Z_q}^*$.
- **Step 3.** Finally, the protocol parameters i.e. $\{G_E, p, q, B, h, PK_{NM}\}$ are produced by the NM.
- **Step 4.** The user U selects its private/public keys as (S_U, PK_U), so that $PK_U = S_U.B$.
- **Step 5.** The server AS selects its private/public keys as (S_{AS}, PK_{AS}), so that $PK_{AS} = S_{AS}.B$.
- **Step 6.** The user U selects its ID_U and computes $\{ID_U.B\}$ as an element of elliptic curve group.

6.2. Registration phase

In the ECCPWS, every new user U must be registered on the NM, so that can use medical services. The registration phase of ECCPWS is done in following steps as depicted in Fig. 4:

- **Step 1.** The user chooses its ID_U, public key PK_U and sends the registration request $\{ID_U.B, PK_U\}$ to the server NM via a secure channel.
- **Step 2.** Once received the request message, the server NM checks out the validity of the message $\{ID_U.B\}$ and picks out a random number $k \in {Z_q}^*$. Next, it computes $Ind_U = k.B$, $U_U = kPK_U$, $x = h(U_U, Ind_U)$ and $v = xS_{NM} + k$. Finally, the server NM sends $\{Ind_U, v\}$ to the user U using the secure channel.
- **Step 3.** After the message $\{Ind_U, v\}$ was received, the U computes $U_U = S_U Ind_U$ and $x = h(U_U, Ind_U)$. Also, the U checks whether $v.B \stackrel{?}{=} xPK_{NM} + Ind_U$. If the user finds any match, the U stores the message $\{Ind_U, v\}$ in its database safely and then computes $K_{UA} = S_U PK_{AS}$, where K_{UA} is the shared secret key between the U and AS. It is obvious that the AS computes this shared secret key as $K_{UA} = S_{AS} PK_U$, which is equal to the computed key at the user side. This is why $K_{UA} = S_U PK_{AS} = S_U S_{AS}.B = S_{AS} S_U.B = S_{AS} PK_U$. Otherwise, it stops the protocol.

6.3. Authentication phase

In order to receive medical services from the AS, the user and AS must be authenticated to each other. The authentication phase of ECCPWS as depicted in Fig. 5, runs as following steps:

- **Step 1.** The user U selects a random number $r \in {Z_q}^*$ and calculates $Q = rPK_{AS}$, $R = r.B$, $c = v + r + S_U$ and $E_U = E_{K_{UA}}(c, U_U, Ind_U, T_1)$. Next, it sends the message of authentication request $\{E_U, R, T_1\}$ to the server AS.
- **Step 2.** After message $\{E_U, R, T_1\}$ was received by the AS at time T_2, the AS checks the timestamp T_1 using inequality $|T_2 - T_1| \leq \Delta T$, if inequality is not established, then the AS rejects the request. Otherwise, the AS approves T_1, goes to the next step and computes $Q' = S_{AS} R$, $K'_{UA} = S_{AS} PK_U$, $D_{K'_{UA}}(E_U) = \{c, U_U, Ind_U, T'_1\}$. Then,

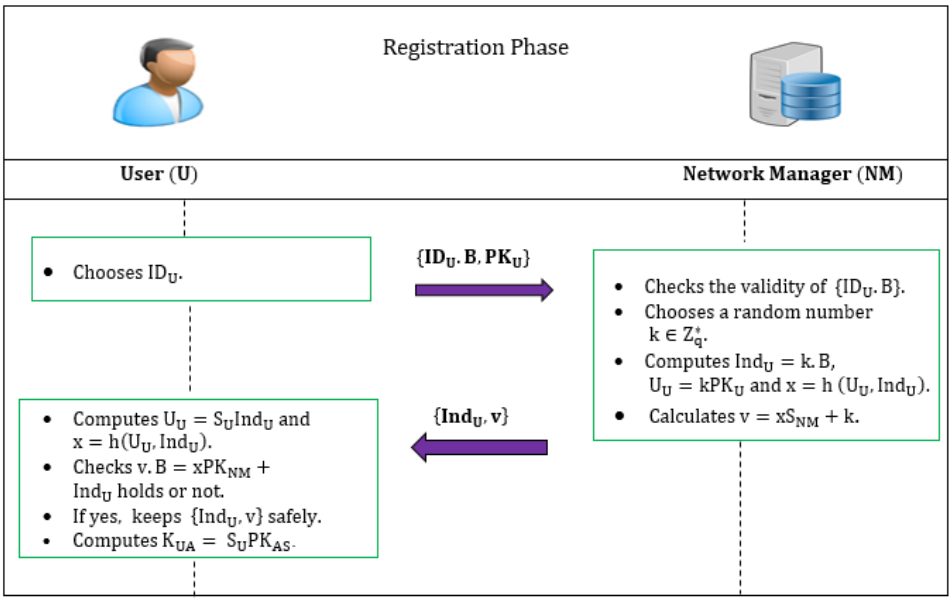

Figure 4: The Registration phase of ECCPWS.

the server AS checks whether $T_1 \stackrel{?}{=} T_1'$ is or not. If it does not hold, the session will end. Otherwise, the AS recognizes the user as a valid user and computes $x = h(U_U, Ind_U)$ and verifies whether $c.B \stackrel{?}{=} xPK_{NM} + R + PK_U + Ind_U$ is or not. If the equation does not hold, the session ends. Otherwise, the AS selects a random number $y \in Z_q^*$ and calculates $Y = y.B$, $Z = yR = yr.B$, $Auth = h(U_U, R, Q', Y, T_3)$ and $SK = h(Auth, U_U, Z)$ as the session key. At last, the server AS sends the $\{Auth, Y, T_3\}$ to the U in response to the authentication request.

- **Step 3.** After the message $\{Auth, Y, T_3\}$ was received by the U at time T_4, the U checks inequality $|T_4 - T_3| \leq \Delta T$, if inequality is not established, then the user rejects the server response. Otherwise, the U continues the process and computes $Auth' = h(U_U, R, Q, Y, T_3)$. Then, it verifies whether $Auth' \stackrel{?}{=} Auth$ is or not. The session ends if it does not hold. Otherwise, the U finds the server AS as a valid server and computes $SK = h(Auth', U_U, rY)$ as the session key. Since $rY = ry.B = yr.B = Z$, it is obvious the generated session key in the U's side is equal to generated session key in the AS's side.

7. Security analysis of ECCPWS

This subsection discusses about the ECCPWS's security. Security of proposed protocol has been excavated informally and formally through Scyther tools [49], ROR model [39, 40], BAN logic [50] and AVISPA tool [51].

7.1. Informal security analysis of ECCPWS

This part informally discusses about security of the proposed scheme. The results of security assessments of ECCPWS compared to other recent related schemes are summarized in Table 2.

7.1.1. The property of perfect secrecy

In the ECCPWS, the user U generates the session key as $SK = h(Auth', U_U, rY)$ in the authentication phase. If the attacker captures the long term secret keys such as K_{UA} and S_{AS}, hence the adversary is not able to calculate the session keys used in the previous sessions. Because the value of the session key depends on the random values y and r, which are related to current session and is not feasible for the adversary to solve ECCDHP to retrieve y and r from Y and R, respectively. So, the ECCPWS has the feature of forward secrecy. Similarly, if the attacker knows the long term secret keys, hence the adversary not able to calculate the session keys used in the future sessions. Therefore, the ECCPWS has the feature of backward secrecy. As a result, according to what mentioned above, it is concluded that the ECCPWS has the feature of perfect secrecy.

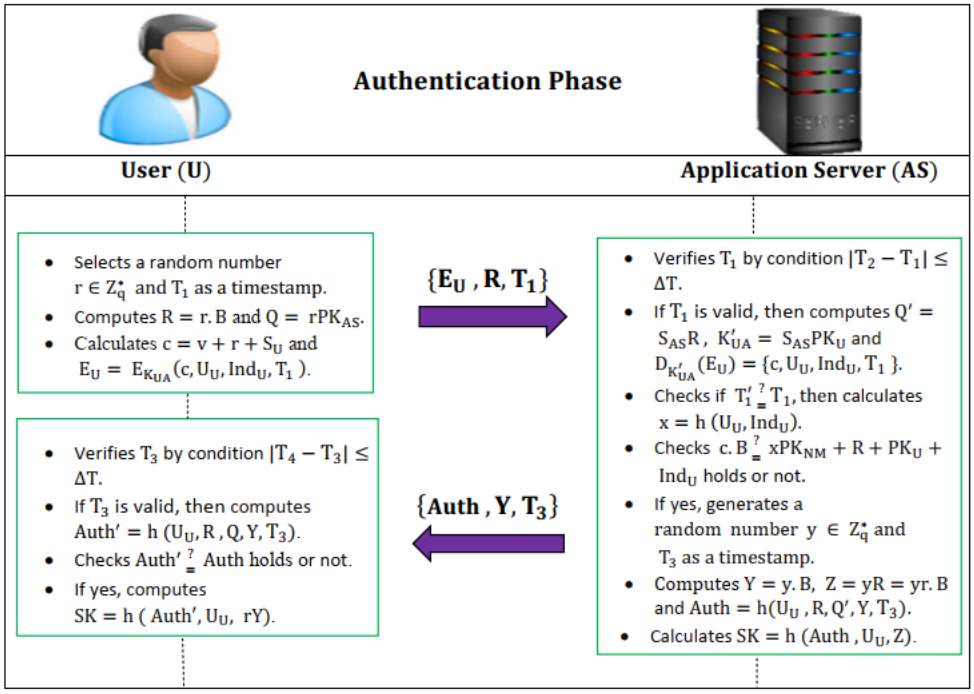

Figure 5: The Authentication phase of ECCPWS.

Table 2
The security features comparison of ECCPWS with ECC-based recent protocols.

Security feature	[34]	[47]	[48]	[20]	ECCPWS
Perfect secrecy	×	×	✓	✓	✓
No key control	×	×	×	✓	✓
Resistance to client (user) impersonation attack	✓	✓	✓	✓	✓
Resistance to NM impersonation attack	✓	✓	✓	×	✓
Mutual authentication between U and NM	×	✓	✓	✓	✓
Untraceability	✓	×	×	×	✓
Resistance to modification or manipulation attack	✓	✓	✓	✓	✓
Resistance to replay attack	✓	×	✓	×	✓
Avoid of clock desynchronization problem	×	✓	✓	×	✓
Resistance to passive insider secret disclosure attack	×	✓	✓	×	✓
Resistance to man-in-the-middle attack	×	✓	✓	×	✓
User anonymity	✓	✓	✓	✓	✓

7.1.2. The property of no key control

In the ECCPWS, the U generates the key used in the current session as $SK = h(Auth', U_U, rY)$. Since, the value of session key depends on the random values y and r, which are selected by AS and U, respectively and SK is protected through Y and R with ECC. So, it can be seen that both the U and AS play same role to produce the session key for every session and before the execution of the protocol neither the U nor AS can compute the session key. Since, it is not feasible for the adversary to solve ECCDHP to retrieve y and r from Y and R, respectively. So, the adversary has no control on the generated session key. Therefore, the ECCPWS provides the property of no key control.

7.1.3. The property of clock synchronization

In the ECCPWS, the U and AS use the random values such as y, r and timestmaps, that these values maintain the freshness of the exchanged messages between participants in each session. Because of the use of the random values and timestamps in exchanged messages, the ECCPWS overcomes desynchronization problem. Therefore, it avoids the problem of clock desynchronization and is resistant to kinds of replay attacks.

7.1.4. The property of mutual authentication

In the ECCPWS, the user validates the registration message $\{Ind_U, v\}$ of the NM by checking whether $v.B \stackrel{?}{=} xPK_{NM} + Ind_U$, where $x = h(U_U, Ind_U)$. The server NM also examines the validity of received ID_U of the user U. Therefore, the proposed scheme provides mutual authentication between the NM and U.

7.1.5. Resistance to impersonation attack

In order to impersonate the AS, the adversary $\mathcal{A}$ must create a valid response for authentication response as $\{Auth, Y, T_3\}$, where $Auth = h(U_U, R, Q', Y, T_3)$. So, the adversary cannot calculate $Auth$ without having the secret values of $Q' = S_{AS}R$, U_U and T_3. Also, U_U is obtained by decryption of E_U using the key of $K'_{UA} = S_{AS}PK_U$ depends to the secret key S_{AS}. Furthermore, to forge the identity of the user U, the adversary $\mathcal{A}$ must generate a valid authentication request $\{E_U, R, T_1\}$, where $R = r.B$, $E_U = E_{K_{UA}}(c, U_U, Ind_U, T_1)$ and $c = v + r + S_U$. The adversary does not have the value of S_U and cannot calculate correct and related c and then E_U. Therefore, the AS can detect the impersonation attack by checking whether $c.B \stackrel{?}{=} xPK_{NM} + R + PK_U + Ind_U$. So, the ECCPWS has complete security against to all kinds of U and AS impersonation attacks.

7.1.6. The property of non-traceability

Since, the exchanged messages on the public channel during the authentication phase are protected using the ECC and hash function, the adversary cannot retrieve any information related to protocol's parties. Therefore, in the ECCPWS, an adversary cannot trace the U and AS by eavesdropping and using exchanged messages on the public and insecure channels.

7.1.7. Resistance to replay attack

As previously mentioned in 7.1.3, if the adversary $\mathcal{A}$ reuses the previous session authentication request $\{E_U, R, T_1\}$, the server AS easily understand that the messages are not fresh. Because, the timestamp is used in calculation of E_U, so the ECCPWS is resistant against all kinds of replay attacks. It worth noting this attack is presented against the Sowjanya *et al.*'s protocol in this paper.

7.1.8. Resistance to modification or manipulation attack

For doing this attack against the proposed protocol, when the user sends $\{E_U, R, T_1\}$ to the AS, if the adversary $\mathcal{A}$ manipulates this message and then sends it to the server, the AS can detect manipulation by checking whether $c.B \stackrel{?}{=} xPK_{NM} + R + PK_U + Ind_U$ is or not. This is why $c = v + r + S_U$ and the secret key of U namely S_U affects the value of c. Also, if the AS sends $\{Auth, Y, T_3\}$ to the user and the adversary changes message $\{Auth, Y, T_3\}$, the U can recognize this change by checking whether $Auth' \stackrel{?}{=} Auth$ is or not. Since $Auth'$ is computed as $Auth' = h(U_U, R, Q, Y, T_3)$, where $U_U = S_U Ind_U$ and it is seen that the secret key of U affects the value of $Auth'$. Therefore, the ECCPWS has full resistance to modification or manipulation attacks.

7.1.9. Resistance to Man-in-the-middle attack

Based on what mentioned in 7.1.8 about the modification attack, the ECCPWS also has full resistance against the man-in-the-middle attack. Because, the integrity of the exchanged messages is checked by each party and if any change occurs, the receiver will understand.

7.1.10. The property of user anonymity

In the proposed protocol, the adversary cannot retrieve any fixed information related to the user U's identity namely U_U and Ind_U. Further, the authentication request $\{E_U, R, T_1\}$ consists of Ind_U and U_U, which are encrypted symmetrically by K_{UA} and the adversary does not know S_U, S_{AS} and K_{UA} for decryption of E_U and faces the ECCDHP for computing K_{UA}. Therefore, it can be seen that the ECCPWS can be able to provide the user anonymity property.

7.1.11. Resistance to passive insider secret disclosure attack

If an insider adversary runs the registration phase i.e. sends its $\{ID_U.B, PK_U\}$ and receives the NM's response i.e. $\{Ind_U, v\}$, s/he cannot obtain the secret value of network manager namely, S_{NM}. Because, for calculating S_{NM}, the adversary needs to know the values of v, k and x. The insider adversary can calculate $U_U = S_U Ind_U$, $x = h(U_U, Ind_U)$ and inverse of x i.e. x^{-1} and receives the value of v from channel. However, s/he cannot obtain the random number k which selected by NM and so, cannot calculate the S_{NM} which equals to $x^{-1}(v-k)$. Therefore, the ECCPWS has full resistance to the passive insider secret disclosure attack. It worth noting this attack is presented against the Sowjanya *et al.*'s protocol in this paper.

7.2. Security analysis with formal methods

There are several methods to evaluate security of protocols. Many of these methods such as ROR model [39, 40], BAN logic [50] and GNY logic [52] are manual formal methods and some of them are automatic such as AVISPA tool [51], Scyther tool [49] and Proverif tool [53]. In this paper, the security analysis of ECCPWS is performed manually through ROR model [39, 40], BAN logic [50] and automatically using Scyther tool [49] and AVISPA tool [51].

7.2.1. *Formal security analysis using Real-Or-Random (ROR) model*

Here, we use widely-accepted ROR model for formal security analysis of the proposed scheme. In this analysis, the main purpose of proof is the security of session key namely SK using the ROR model. Hence, this proof is stated in Theorem 1. The ROR model considers the following components:

1. **Adversary:** In the Dolev-Yao model [46], there are assumptions that all communications can be controlled by $\mathcal{A}$ and the adversary has complete control over the channel. Also in this adversary model, the adversary has abilities such as eavesdropping, modifying, deleting the exchanged messages, fabricating new messages and injecting messages between two entities during the communication. So, $\mathcal{A}$ has access to following queries:
   - $Execute(\mathcal{O}^t, \mathcal{O}^u, \mathcal{O}^v)$: The adversary $\mathcal{A}$ executes the query to oracles NM, AS and U at times u, v and t, respectively for eavesdropping and getting the transmitted messages among three legitimate participants U, NM and AS. Also, passive attacks are modeled under this query.
   - $Reveal(\mathcal{O}^t)$: In this query, the session key i.e. SK created by $\mathcal{O}^t$ (and its partner) is disclosed for $\mathcal{A}$ in the current session.
   - $Test(\mathcal{O}^t)$: In this query, the security of the session key SK between the U and AS has been modeled. So, an unbiased coin a is selected at the beginning of the experiment and its result only is known for $\mathcal{A}$. Based on output, the decision makes. Assume that the adversary $\mathcal{A}$ runs this query and then $\mathcal{O}^t$ returns SK in case $a = 1$ or returns a random number if $a = 0$; otherwise, it returns a null value ($\perp$).
   - $Send(\mathcal{O}^t, msg)$: This query is used to send a message msg to a participant $\mathcal{O}^t$ or receive the response of message from $\mathcal{O}^t$ with $\mathcal{A}$.
2. **Participants:** The oracles of $\mathcal{O}^t_U$, $\mathcal{O}^u_{NM}$ and $\mathcal{O}^v_{AS}$ are used for participants U, NM and AS at time t, u and v, respectively.
3. **Accepted state:** The oracles of $\mathcal{O}^{t_1}$ and $\mathcal{O}^{t_2}$ go to accepted state, after receiving the last protocol message and successful authentication to each other. Also, the oracles consider the session identification sid for the current session.
4. **Partnering:** Two oracles of $\mathcal{O}^{t_1}$ and $\mathcal{O}^{t_2}$ are partners to each other, if we have three conditions such as: 1) Oracles $\mathcal{O}^{t_1}$ and $\mathcal{O}^{t_2}$ be in accepted states. 2) Oracles $\mathcal{O}^{t_1}$ and $\mathcal{O}^{t_2}$ mutually authenticate each other and share the same sid and 3) Oracles $\mathcal{O}^{t_1}$ and $\mathcal{O}^{t_2}$ be mutual partners to each other.
5. **Freshness:** If the session key SK between the U and AS has been not revealed by the adversary $\mathcal{A}$ using the $Reveal(\mathcal{O}^t)$ query, then we say that $\mathcal{O}^t_U$ or $\mathcal{O}^v_{AS}$ is fresh.
6. **Semantic security of the session key:** The semantic security of SK among the U and AS has been specified based on the indistinguishability property between the real session key and a random value, which is selected by $\mathcal{A}$. The adversary can produce several $Test$ queries for oracles $\mathcal{O}^t_U$, $\mathcal{O}^v_{AS}$ and $\mathcal{O}^u_{NM}$. The output of $Test$ query should be related to the random bit a. At the end of experiment, the adversary $\mathcal{A}$ guesses a bit a'. If the condition $a' = a$ is met, the adversary can win the game. Given $Succ$ is an event that the $\mathcal{A}$ wins the game, the advantage of $\mathcal{A}$ in breaking the security of ECCPWS is defined by:
   $$Adv_{\mathcal{A}}^{ECCPWS} = |2.Pr[Succ_{\mathcal{A}}^{Game_i}] - 1|$$
   The ECCPWS is secure, if $Adv_{\mathcal{A}}^{ECCPWS} \leq \epsilon$ for negligible values $\epsilon > 0$.
7. **Random oracle:** The ECCPWS uses a collision-resistant one-way hash function $H(.)$. This function is modeled as a random oracle in the form of $H(.)$. It is assumed that the adversary $\mathcal{A}$ and all of the participants can access to $H(.)$.

Security Proof: The security of SK in the ECCPWS is proved based on Theorem 1 under the ROR model.

Theorem 1: Suppose that the adversary $\mathcal{A}$ uses polynomial time t in the ROR model, and q_{hash}, $|Hash|$ and $Adv_{\mathcal{A}}^{ECDDHP}(t)$ show the number of hash queries, range space of hash function $H(.)$ and advantage of $\mathcal{A}$ in breaking $ECDDHP$, respectively. So, the advantage of $\mathcal{A}$ in breaking security of ECCPWS to get SK among the U and AS during the authentication phase can be estimated as:

$$Adv_{\mathcal{A}}^{ECCPWS}(t) \leq \frac{q_{hash}^2}{|Hash|} + 2.Adv_{\mathcal{A}}^{ECDDHP}(t)$$

Proof 1: This proof includes of three games in the form of $Game_i$ with $(i = 0, 1, 2)$. $Succ_{\mathcal{A}}^{Game_i}$ is an event that the $\mathcal{A}$ can estimate the random bit a of a flipped unbiased coin in the $Game_i$ using the $Test$ query successfully. It can be shown that the advantage of $\mathcal{A}$ in winning the $Game_i$ is expressed as $Adv_{\mathcal{A},Game_i}^{ECCPWS} = Pr[Succ_{\mathcal{A}}^{Game_i}]$. The details of games are explained below in sequel.

- $Game_0$: In this game as the initial game, the adversary $\mathcal{A}$ implements the "real attack" on the ECCPWS under the ROR model. At the beginning of

$Game_0$, the bit a is chosen randomly. So, this result follows that:

$$Adv_{A}^{ECCPWS}(t) = |2.Adv_{A,Game_0}^{ECCPWS} - 1| \quad (1)$$

- $Game_1$: This game simulates an "eavesdropping attack" performed by the eavesdropper $\mathcal{A}$, which the adversary $\mathcal{A}$ can execute the $Execute(\mathcal{O}^t, \mathcal{O}^u, \mathcal{O}^v)$ query. $\mathcal{A}$ can eavesdrop all exchanged messages $\{E_U, R, T_1\}$ and $\{Auth, Y, T_3\}$ during authentication phase using the $Execute$ query. The adversary $\mathcal{A}$ needs to execute the $Reveal$ and $Test$ queries at the end of the game. Then, the adversary $\mathcal{A}$ recognizes whether output of the $Test$ query is the real SK or random value. Note that the session key is computed as $SK = h(Auth, U_U, Z)$. Hence, the security of session key SK depends on the long term secret values S_{AS}, S_U and random values r and y. Therefore, eavesdropping of exchanged messages $\{E_U, R, T_1\}$ and $\{Auth, Y, T_3\}$ does not increase the probability of winning the adversary $\mathcal{A}$ in the obtaining of SK in $Game_1$ without these secret values. It is clear that $Game_0$ and $Game_1$ are indistinguishable, namely;

$$Pr[Succ_{\mathcal{A}}^{Game_0}] = Pr[Succ_{\mathcal{A}}^{Game_1}] \quad (2)$$

Also, $\mathcal{A}$ requires the random numbers (k, r, y). These values are not known to $\mathcal{A}$. It follows that:

$$Adv_{\mathcal{A},Game_0}^{ECCPWS} = Adv_{\mathcal{A},Game_1}^{ECCPWS} \quad (3)$$

- $Game_2$: In this game, the simulation of the $Hash$ query has been modeled. The adversary $\mathcal{A}$ tries to make several $Hash$ queries to find a collision in output of hash functions. When the $Hash$ query is executed by $\mathcal{A}$, no collision occurs. In the message $\{Auth, Y, T_3\}$, the term $Auth$ is safeguarded by the "collision-resistant one-way hash function $H(.)$". Due to the ideal assumption of the hash function, no collision occurs if the $Hash$ query is executed by $\mathcal{A}$. Again, in the intercepted messages $R = r.B$ and $Y = y.B$, there is "computationally infeasible problem" for the adversary $\mathcal{A}$ to derive $Z = yr.B$ due to intractability of $ECDDHP$. Hence, to derive SK between the U and AS, the adversary needs Z, which retrieving Z is difficult for $\mathcal{A}$ to solve $ECDDHP$ in polynomial time t and also there is no collision in the output of hash function in the $Auth$. It is worth noting that both the $Game_1$ and $Game_2$ are " indistinguishable" except the simulation of the $Hash$ query in $Game_2$. So, we get the following result using the results obtained from the intractability of $ECDDHP$ and the birthday paradox:

$$\begin{aligned} &|Adv_{\mathcal{A},Game_1}^{ECCPWS} - Adv_{\mathcal{A},Game_2}^{ECCPWS}| \\ &\leq \frac{q_{hash}^2}{2|Hash|} + Adv_{\mathcal{A}}^{ECDDHP}(t) \end{aligned} \quad (4)$$

The adversary $\mathcal{A}$ simulates all the queries and wins the game only with guessing the bit a once and execution of $Test$ and $Reveal$ queries. Then, we have

$$Adv_{\mathcal{A},Game_2}^{ECCPWS} = \frac{1}{2} \quad (5)$$

Therefore, based on equations (1) and (2) we have:

$$\begin{aligned} \frac{1}{2}Adv_{\mathcal{A}}^{ECCPWS}(t) &= |Adv_{\mathcal{A},Game_0}^{ECCPWS} - \frac{1}{2}| \\ &= |Adv_{\mathcal{A},Game_1}^{ECCPWS} - \frac{1}{2}| \end{aligned} \quad (6)$$

Based on Equations (3), (4), (5) and (6), we have:

$$\begin{aligned} &\frac{1}{2}Adv_{\mathcal{A}}^{ECCPWS}(t) \\ &= |Adv_{\mathcal{A},Game_1}^{ECCPWS} - Adv_{\mathcal{A},Game_2}^{ECCPWS}| \\ &\leq \frac{q_{hash}^2}{2|Hash|} + Adv_{\mathcal{A}}^{ECDDHP}(t) \end{aligned} \quad (7)$$

As a result, we multiply both sides of Equation (7) by a factor of 2. So, the final result, which is theorem 1, is proved as below:

$$Adv_{\mathcal{A}}^{ECCPWS}(t) \leq \frac{q_{hash}^2}{|Hash|} + 2.Adv_{\mathcal{A}}^{ECDDHP}(t).$$

This inequality shows the advantage of $\mathcal{A}$ in breaking the security of ECCPWS to obtain SK between the U and AS in the authentication phase. Thus, the SK security in the ECCPWS is proved based on the ROR model.

7.2.2. Formal security analysis using the Scyther tool

Scyther [49] is an impressive automatic formal tool for analysis of security schemes to recognize attacks. This in its standard version works based on the Dolev-Yao model [46]. Hence, it investigates security of protocols by exploiting of security claims. As well as, it examines all types of security claims in the protocol and produces a graph for any attack according to each claim. The Scyther's language is Security Protocols Description Language (SPDL). This tool provides a graphical user inter-

Claim				Status	Comments
proposed	U	Proposed, U2	Secret r	OK	No attacks within bounds.
		Proposed, U3	Secret idu	OK	No attacks within bounds.
		Proposed, U4	Secret B	OK	No attacks within bounds.
		Proposed, U5	Secret t	OK	No attacks within bounds.
		Proposed, U6	Secret y	OK	No attacks within bounds.
		Proposed, U7	Alive	OK	No attacks within bounds.
		Proposed, U8	Weakagree	OK	No attacks within bounds.
		Proposed, U9	Niagree	OK	No attacks within bounds.
		Proposed, U10	Nisynch	OK	No attacks within bounds.
	NM	Proposed, NM1	Secret t	OK	No attacks within bounds.
		Proposed, NM2	Secret idu	OK	No attacks within bounds.
		Proposed, NM3	Secret B	OK	No attacks within bounds.
		Proposed, NM4	Alive	OK	No attacks within bounds.
		Proposed, NM5	Weakagree	OK	No attacks within bounds.
		Proposed, NM6	Niagree	OK	No attacks within bounds.
		Proposed, NM7	Nisynch	OK	No attacks within bounds.
	AS	Proposed, AS1	Secret B	OK	No attacks within bounds.
		Proposed, AS2	Secret idu	OK	No attacks within bounds.
		Proposed, AS3	Secret y	OK	No attacks within bounds.
		Proposed, AS4	Secret t	OK	No attacks within bounds.

Figure 6: The verification results of security claims of the U in ECCPWS using the Scyther tool.

Claim				Status	Comments
proposed	U	Proposed, U1	Secret r	OK	No attacks within bounds.
		Proposed, U2	Secret idu	OK	No attacks within bounds.
		Proposed, U3	Secret B	OK	No attacks within bounds.
		Proposed, U4	Secret t	OK	No attacks within bounds.
		Proposed, U5	Secret y	OK	No attacks within bounds.
		Proposed, U6	Alive	OK	No attacks within bounds.
		Proposed, U7	Weakagree	OK	No attacks within bounds.
		Proposed, U8	Niagree	OK	No attacks within bounds.
		Proposed, U9	Nisynch	OK	No attacks within bounds.
	NM	Proposed, NM2	Secret t	OK	No attacks within bounds.
		Proposed, NM3	Secret idu	OK	No attacks within bounds.
		Proposed, NM4	Secret B	OK	No attacks within bounds.
		Proposed, NM5	Alive	OK	No attacks within bounds.
		Proposed, NM6	Weakagree	OK	No attacks within bounds.
		Proposed, NM7	Niagree	OK	No attacks within bounds.
		Proposed, NM8	Nisynch	OK	No attacks within bounds.
	AS	Proposed, AS1	Secret B	OK	No attacks within bounds.
		Proposed, AS2	Secret idu	OK	No attacks within bounds.
		Proposed, AS3	Secret y	OK	No attacks within bounds.
		Proposed, AS3	Secret t	OK	No attacks within bounds.

Figure 7: The verification results of security claims of the NM in ECCPWS using the Scyther tool.

face. Scyther is a tool for the formal verification of security protocols according to complete cryptography assumption, i.e. it assumes every cryptography function which used is perfectly secure. Also, the Scyther assumes an adversary can retrieve the exchanged messages, if s/he has decryption key. This tool investigates secrecy and authentication in the security protocols. There are two very important assumptions in the model of this tool. The first assumption is that, its cryptographic model is black box, and the second assumption is that the messages sent are descent. Therefore, in this tool, protocols are modeled based on role definition. There are many security claims such as $Alive$, $Nisynch$, $secret$, $weakagree$ and etc. in the Scyther tool. For example, $Alive$ refers to the claim that ensures an intended communication party R has executed a set of events. $Nisynch$ is a claim that ensures all exchanged messages have been sent by sender and the receiver has received all the sent messages. $Claim(R; secret; rt)$ implies that R claims that rt should be unknown to the attacker. $weakagree$ is used to guarantee the robustness of the protocol against impersonation attacks. After specifying the security claims and roles, the security verification of the protocol begins with the execution of the verify command. It can be seen that the output

Claim				Status	Comments
proposed	U	Proposed, U1	Secret r	OK	No attacks within bounds.
		Proposed, U2	Secret idu	OK	No attacks within bounds.
		Proposed, U3	Secret B	OK	No attacks within bounds.
		Proposed, U4	Secret t	OK	No attacks within bounds.
		Proposed, U5	Secret y	OK	No attacks within bounds.
		Proposed, U6	Alive	OK	No attacks within bounds.
		Proposed, U7	Weakagree	OK	No attacks within bounds.
		Proposed, U8	Niagree	OK	No attacks within bounds.
		Proposed, U9	Nisynch	OK	No attacks within bounds.
	NM	Proposed, NM1	Secret t	OK	No attacks within bounds.
		Proposed, NM2	Secret idu	OK	No attacks within bounds.
		Proposed, NM3	Secret B	OK	No attacks within bounds.
		Proposed, NM4	Alive	OK	No attacks within bounds.
		Proposed, NM5	Weakagree	OK	No attacks within bounds.
		Proposed, NM6	Niagree	OK	No attacks within bounds.
		Proposed, NM7	Nisynch	OK	No attacks within bounds.
	AS	Proposed, AS2	Secret B	OK	No attacks within bounds.
		Proposed, AS3	Secret idu	OK	No attacks within bounds.
		Proposed, AS4	Secret y	OK	No attacks within bounds.
		Proposed, AS5	Secret t	OK	No attacks within bounds.

Figure 8: The verification results of security claims of the AS in ECCPWS using the Scyther tool.

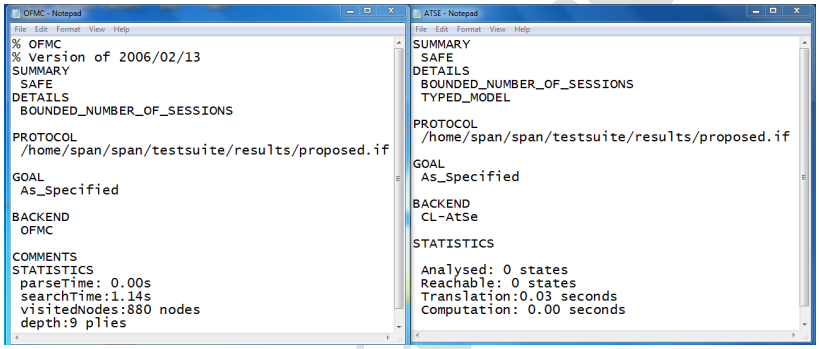

Figure 9: The verification results of ECCPWS using the OFMC and ATSE back-ends.

of Scyther tool consists of two modes: The first mode is when an attack against the protocol is detected, that the graphical scenario of the detected attack is also specified. The second mode is when the protocol is recognized secure by this tool and its security claims are confirmed. Given the Scyther tool, both the correctness and authenticity of the security protocols can be examined. Therefore, we modeled our proposed protocol using the SPDL to verify protocol's entities claims. The SPDL implementation of ECCPWS is represented in Appendix A. It can be seen the proposed protocol is modeled based on the definition of the role of participants in the protocol such as NM, AS and U and then these roles communicate to each other through *recv* and *send* channels. As well as, its verification results have been shown in Fig. 6, Fig. 7 and Fig. 8, respectively, which show that there are no weaknesses in the ECCPWS.

7.2.3. *Formal security verification using the AVISPA tool*

In this section, we first briefly introduce the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [51]. Then, we implement our proposed protocol using the High-Level Protocol Specification Language (HLPSL) in this tool. The AVISPA tool is a push-button tool for the automated formal security verification of protocols, which shows whether the authentication protocol is SAFE or UNSAFE against different attacks. The architecture of AVISPA tool is depicted in [53]. In this tool, there are four back-ends, which are explained as below:

- "On-the-Fly Model Checker (OFMC)": This back-end confirms or disapproves the protocol by searching the transition system described by the IF specification.
- "Constraint-Logic-based Attack Searcher (CL-

AtSe)": This back-end does both protocol falsification and verification for bounded numbers of sessions. CL-AtSe uses algebraic properties of cryptographic operators and several kinds of optimizations to reduce and eliminate redundancies in the protocol symbolic execution. Also, in this back-end, the intruder messages are saved using variables.

- "SAT-based Model Checker (SATMC)": It considers the protocol model and performs both protocol falsification and verification. First, the initial state, the set of states representing violation of the security properties and transition interface specified by the IF create a command. Then, this command fed to a SAT solver and as a result, any model found is translated into an attack.

- "Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP)" : This back-end approximates the adversary knowledge via tree structure.

Given the protocol has been wrote in HLPSL, the HLPSL code as input to each of four back-ends is translated into the "Intermediate Format (IF)" using HLPSL2IF translator. IF is a low-level language that, prepares the model for analysis using four back-ends. Then, IF creates the "output format (OF)". The different parts of OF are described below:

- DETAILS: It says "the explanation of why the tested protocol is resulted as safe, had an attack or why the analysis is inconclusive".

- SUMMARY: It states "whether the protocol is unsafe, safe or the analysis is inconclusive".

- GOAL: It explains "the goals of analysis performed by AVISPA tool".

- BACK-END: It prepares "the name of the back-end that is used for the analysis, such as OFMC, CL-AtSe, SATMC and TA4SP".

- PROTOCOL: It explains "the used HLPSL specification of protocol in IF".

- VULNERABILITY: It is including "the trace of vulnerability and relevant comments in the security of protocol".

In the HLPSL implementation of ECCPWS, the information will be obtained from the specification of protocol. In the HLPSL, we have composition roles and main roles. The main roles show different agents in the protocol (i.e. the roles of the user U, the network manager NM and the application server AS in the ECCPWS). The composition roles consist of session, goal and environment, that show different scenarios involving main roles. The specification of main roles U, NM and AS of ECCPWS in HLPSL are shown in Appendix B. Also, the composition roles specification of session, environment and goal are shown in Appendix B. The roles of environment, session and goal include the adversary knowledge, all the constants used, goals, specification of sessions and all parameters of agents. In order to do the formal security analysis of ECCPWS, we have used the "Security Protocol ANimator for AVISPA (SPAN)". The simulation outcomes of ECCPWS under the OFMC and ATSE back-ends have been shown in Fig. 9. It ensures that the proposed scheme is secure against the different attacks such as the replay and man-in-the-middle attacks.

7.2.4. Security analysis using the BAN logic

Burrows–Abadi–Needham logic or in brief BAN logic [50] is one of the formal logic procedures for analysis of security protocols. BAN logic includes the set of rules for describing and analyzing the various security protocols [50]. In this logic, a set of assumptions and rules are used to prove the mutual authentication of the protocol parties. This logic is constructed on the belief and honesty of entities. Particularly, the BAN logic aids users to characterize, whether exchanged messages are reliable or not. A generic BAN logic contains three tests:

- Verification of message source

- Verification of source trustworthiness

- Verification of message freshness

Inference Rules: The different sets of inference rules (IR) of BAN logic using represented notations in Table 3 are listed in Table 4.

Since the channel used in the registration phase is secure, it is clear that all of security goals through BAN logic are accessible. Therefore, in this subsection, the BAN logic is only used to prove the security of authentication phase of ECCPWS. The BAN logic security proof of authentication phase of ECCPWS is performed as below:

- **Generic form of messages**: The exchanged messages during the authentication phase of ECCPWS can be expressed as follows: **Message 1:** $U_i \longmapsto AS : \{E_U, R, T_1\} : \left\{\{c, U_U, Ind_U, T_1\}_{K_{UA}}, r.B, T_1\right\}$ **Message 2:** $AS \longmapsto U_i : \{Auth, Y, T_3\} : \left\{h(U_U, R, Q', Y, T_3), y.B, T_3\right\}$

Table 3
Some of used notations in BAN logic.

Notation	Description
$P \lhd X$	The agent P sees the message X.
$P\mid \equiv X$	The agent P believes the message X.
$P\mid \Rightarrow X$	P has jurisdiction over the message X, namely, P understands if the message changes.
$\sharp X$	X is a fresh message.
$\{X\}_K$	The message X is encrypted using the key K.
$(X)_h$	The hash of message X.
$< X >_Y$	Formula X is combined with formula Y.
(X,Y)	X or Y is a part of formula (X,Y).
SK	The session key.
$\mathrm{P} \overset{K}{\rightleftharpoons} \mathrm{Q}$	K is a shared secret key between P and Q.
$P \overset{K}{\leftrightarrow} \mathrm{Q}$	Principals P and Q communicate to each other via shared key K.
$P\mid \sim X$	Principal P said once the message X.
$\overset{K^+}{\longmapsto} \mathrm{P}$	Public/private keys of P are K^+ and K^-, respectively.

Table 4
The used BAN logic rules in the security proof of ECCPWS.

Notation	Description
$\frac{P\mid\equiv\sharp(X)}{P\mid\equiv\sharp(X,Y)}$	IR1: This rule ensures that entire formula is believed to be fresh if one part of it is believed to be fresh. So, it means if the agent P believes that the message X is fresh, then it is deduced that the agent P believes that the message (X,Y) is fresh.
$\frac{P\mid\equiv X, P\mid\equiv Y}{P\mid\equiv(X,Y)}$	IR2: This means, if the agent P believes the messages X and Y distinctly, then P believes the combined formula (X,Y).
$\frac{P\mid\equiv(X,Y)}{P\mid\equiv X}$	IR2*: This means, if the agent P believes the combined formula (X,Y), then believes the message X or Y.
$\frac{P\mid\equiv Q\mid\Rightarrow X, P\mid\equiv Q\mid\equiv X}{P\mid\equiv X}$	IR3: This rule used to validate the control of an entity on message. It means, if agent P believes Q has jurisdiction over X and P believes that the agent Q believes the message X, then P believes the X.
$\frac{P\mid\equiv\sharp(X), P\mid\equiv Q\mid\sim X}{P\mid\equiv Q\mid\equiv X}$	IR4: This rule used to check freshness of message and subsequently the sender's belief to freshness of it. So, it means, if the agent P believes the message X is fresh and P believes the agent Q said X once, then the agent P believes Q believes X.
$\frac{P\mid\equiv \overset{K}{\rightleftharpoons} Q, P\lhd\{X\}_K}{P\mid\equiv Q\mid\sim X}$	IR5: This rule used for interpretation of encrypted messages. Therefore, it means, if the agent P believes the secret K to be shared between itself and the agent Q and also P sees that the message X is encrypted with the secret K, then it is deduced that the agent P believes that the agent Q once said the message X.
$\frac{P\mid\equiv \overset{K^+}{\longmapsto} Q, P\lhd\{X\}_{K^-}}{P\mid\equiv Q\mid\sim X}$	IR5*: This rule used for interpretation of encrypted messages. Therefore, it means if the agent P believes K^+ is public key of the agent Q and also P sees that the message X is encrypted with its secret key K^-, then it is deduced that the agent P believes that the agent Q once said X.
$\frac{P\mid\equiv Q\mid\sim(X,Y)}{P\mid\equiv Q\mid\sim X}$	IR6: It means that if P believes the agent Q said (X,Y), then it is deduced that P also believes the agent Q once said X.
$\frac{P\mid\equiv X}{P\mid\equiv(X)_h}$	IR7: It means that if the agent P believes the message X, then it is deduced that the agent P believes the hash value of X, namely $h(X)$.

- **Determination of the Protocol as Relevant and Idealized form of BAN Logic**: In this step, we convert the exchanged messages in the authentication phase of ECCPWS to the BAN logic relevant form. Here, the exchanged messages are idealized. This means, sent messages in unencrypted form that do not have any security enhancements, are deleted. So, the idealized form of messages in the authentication phase of ECCPWS is as follows where K^{-AS} shows the private key of AS, while PK_{AS} shows its public key and K_{UA} shows the shared symmetric key between the U and AS.

$$IM1 : AS \lhd \{c, U_U, Ind_U, T_1\}_{K_{UA}}, r.B$$
$$IM2 : U_i \lhd \{U_U, R, Y, T_3\}_{K^{-AS}}, y.B$$

- **Determination of Initial Assumptions**: For exam-

Table 5
Security assumptions and goals in the security proof of ECCPWS with BAN logic.

Notation	Description
$U_i \mid \equiv r$	A1: U_i believes r.
$U_i \mid \equiv R$	A2: U_i believes R.
$AS \mid \equiv y$	A3: AS believes y.
$AS \mid \equiv Y$	A4: AS believes Y.
$AS \mid \equiv \mathrm{U}_i \mid \Rightarrow U_U$	A5: AS believes that the U_i has jurisdiction over the U_U, namely, U_i understands if U_U changes.
$AS \mid \equiv \mathrm{U}_i \mid \Rightarrow Ind_U$	A6: AS believes that the U_i has jurisdiction over the Ind_U, namely, U_i understands if Ind_U changes.
$U_i \mid \equiv \mathrm{U}_i \overset{K_{UA}}{\rightleftharpoons} \mathrm{A}S$	A7: U_i believes that the K_{UA} is a shared secret key between the U_i and AS.
$AS \mid \equiv \mathrm{A}S \overset{K_{UA}}{\rightleftharpoons} \mathrm{U}_i$	A8: AS believes that the K_{UA} is a shared secret key between the AS and U_i.
$U_i \mid \equiv \overset{PK_{AS}}{\mapsto} \mathrm{AS}$	A9: U_i believes that PK_{AS} is the public key of AS.
$U_i \mid \equiv \sharp r$	A10: U_i believes that the r is fresh.
$U_i \mid \equiv AS \mid \Rightarrow Y$	A11: U_i believes that the AS has jurisdictions over the Y, namely, AS understands if Y changes.
$U_i \mid \equiv U_U$	A12: U_i believes U_U.
$AS \mid \equiv U_i \mid \Rightarrow R$	A13: AS believes that the U_i has jurisdictions over the R, namely, U_i understands if R changes.
$AS \mid \equiv S_{AS}$	A14: AS believes S_{AS}.
$AS \mid \equiv \sharp U_U$	A15: AS believes that the U_U is fresh.
$AS \mid \equiv \sharp r$	A16: AS believes that the r is fresh.
$AS \mid \equiv U_i \mid \equiv R$	A17: AS believes that the U_i believes R.
$U_i \mid \equiv \sharp T_1$	A18: U_i believes that the T_1 is fresh.
$U_i \mid \equiv \sharp T_2$	A19: U_i believes that the T_2 is fresh.
$AS \mid \equiv \sharp T_3$	A20: AS believes that the T_3 is fresh.
$AS \mid \equiv \sharp T_4$	A21: AS believes that the T_4 is fresh.
$AS \mid \equiv SK$	Goal1: AS believes the shared secret key SK.
$U_i \mid \equiv SK$	Goal2: U_i believes the shared secret key SK.

ining the security feature of mutual authentication, many assumptions are made to begin the state of the process. The BAN logic assumptions of ECCPWS are represented in Table 5.

- **Definition of the Security Goals**: In order to examine the security of ECCPWS, we should define its security goals. According to the analytic procedure of the BAN logic, our scheme meets two goals, which are stated in Table 5.

- **Achievement of Security Goals**: In this step, by applying the BAN logic rules, which are represented in Table 4, to the ECCPWS's idealized messages and assumptions, its security goals are deduced in Table 6. It worth noting that the BAN's logic rules are written as a fraction A/B, which means that if A exists, then B is true. In proving the security goals, the protocol's idealized messages and assumptions are used to make the numerator of BAN's logic rules, and then it is concluded that the denominator of those rules are valid.

Therefore, the security goal "Goal1" and "Goal2" will be achieved, which shows the server AS and user U_i believe its computed shared secret key. Based on the above proofs, it is inferred that the ECCPWS gains every two goals. "Goal1" means that the AS believes the shared secret key, namely SK, and "Goal2" means that the U_i believes the shared secret key, namely, SK. As a result, it is demonstrated that this scheme is secure according to the BAN logic. In this protocol, the security is created with secrecy of the session key namely, SK. In other words, if the session key SK cannot be computed with the attacker, our suggested protocol is secure. The above analysis proves that the suggested scheme has complete security and can achieve mutual authentication.

8. Comparative Analysis

Here, we compare the proposed scheme with other similar recently proposed schemes in terms of performance features such as storage cost, communication cost and computational cost. The results of these comparisons have been shown in Tables 7, 10 and 11, respectively.

8.1. Storage cost comparison

In order to compare the storage cost in various protocols, we used different parameters length that represented in Table 8.

In the Sowjanya *et al.*'s protocol [20], the user stores the message $\{Ind_U, right, v\}$. Ind_U is a point in elliptic curve group with 1024 bits, parameter $right$ with 64 bits, $v = xS_{NM}$ that $(h, x, S_{NM}, S_U) \in Z_q{}^*$ with 160 bits. So, these parameters occupy $(1024 + 64 + 160 = 1248)$ bits from the capacity of memory. Also, the AS stores only its secret key, namely S_{AS} with 160 bits. Therefore, the total storage cost of this scheme is 1408 bits.

In the Rostampour *et al.*'s protocol [48], the user saves $\{CK_i', PID_i\}$ with 1024 and 160 bits, respectively, which

Table 6
The proof of security ECCPWS with BAN logic.

Message	Assumptions	Inference Rules	Postulate	Security Goal
$IM1$	$A8$	$IR5$	$P1 : AS\| \equiv U_i\| \sim \{c, U_U, Ind_U, T_1\}$	–
$P1$	–	$IR6$	$P2 : AS\| \equiv U_i\| \sim c$	–
$P1$	–	$IR6$	$P3 : AS\| \equiv U_i\| \sim U_U$	–
$P1$	–	$IR6$	$P4 : AS\| \equiv U_i\| \sim Ind_U$	–
$P1$	–	$IR6$	$P5 : AS\| \equiv U_i\| \sim T_1$	–
$P2$	–	$IR6$	$P6 : AS\| \equiv U_i\| \sim r$	–
$P6$	$A13, A17$	$IR3, IR4$	$P7 : AS\| \equiv R$	–
$P7$	$A14$	–	$P8 : AS\| \equiv RS_{AS} = Q'$	–
$P3$	$A15$	$IR4$	$P9 : AS\| \equiv U_i\| \equiv U_U$	–
$P9$	$A5$	$IR3$	$P10 : AS\| \equiv U_U$	
$P7, P8, P10$	$A4, A17, A21$	$IR2$	$P11 : AS\| \equiv (U_U, R, Q', Y, T_3)$	–
$P11$	–	$IR7$	$P12 : AS\| \equiv (U_U, R, Q', Y, T_3)_h = Auth$	–
$P7$	$A3$	–	$P13 : AS\| \equiv yR = Z$	–
$P10, P12, P13$	–	$IR2$	$P14 : AS\| \equiv (Auth, U_U, Z)$	–
$P14$	–	$IR7$	$P15 : AS\| \equiv (Auth, U_U, Z)_h = SK$	$Goal1$
$IM2$	$A9$	$IR5^*$	$P16 : U_i\| \equiv AS\| \sim (U_U, R, Y, T_3)$	–
–	$A10$	$IR1$	$P17 : U_i\| \equiv \sharp(U_U, R, Y, T_3)$	–
$P16, P17$	–	$IR4$	$P18 : U_i\| \equiv AS\| \equiv (U_U, R, Y, T_3)$	–
$P18$	–	$IR6$	$P19 : U_i\| \equiv AS\| \equiv U_U$	–
$P18$	–	$IR6$	$P20 : U_i\| \equiv AS\| \equiv R$	–
$P18$	–	$IR6$	$P21 : U_i\| \equiv AS\| \equiv Y$	–
$P18$	–	$IR6$	$P22 : U_i\| \equiv AS\| \equiv T_3$	–
$P21$	$A11$	$IR3$	$P23 : U_i\| \equiv Y$	–
–	$A1, A9$	–	$P24 : U_i\| \equiv rPK_{AS} = Q$	–
$P23, P24$	$A12, A2, A21$	$IR2$	$P25 : U_i\| \equiv (U_U, R, Q, Y, T_3)$	–
$P25$	–	$IR7$	$P26 : U_i\| \equiv (U_U, R, Q, Y, T_3)_h = Auth'$	–
$P23$	$A1$	–	$P27 : U_i\| \equiv rY$	–
$P26, P27$	$A12$	$IR2$	$P28 : U_i\| \equiv (Auth', U_U, rY)$	–
$P28$	–	$IR7$	$P29 : U_i\| \equiv (Auth', U_U, rY)_h = SK$	$Goal2$

in overall equals to (1024 + 160 = 1184) bits. Also, the server side stores PID_i, ET as timestamp and CK_i, which in overall equals to (160 + 32 + 160 = 352) bits. Therefore, the total storage cost of the Rostampour *et al.*'s protocol is 1536 bits.

The user and server store the values CID_i, CK' with (160 + 1024 = 1184) bits and $\{CID_i, t', a', e_t', w_{bit-i}\}$ with (160 + 160 + 160 + 160 + 1 = 641) bits in the Kumar *et al.*'s scheme [47], respectively. So, the total storage cost of Kumar *et al.*'s scheme equals to 1825 bits.

In the Li *et al.*'s protocol [34], the user saves $\{Ind_U, E_1, U, k\}$, which occupy (1024 + (1024 + 1024 + 64) + 1024 + 160 = 4320) bits. Also, the application server stores two secret keys of S_{AS} and S_{NA}, which occupy (160 + 160 = 320) bits. So, the total storage cost of Li *et al.*'s protocol equals to (4320 + 320 = 4640) bits.

In the ECCPWS, the user stores $\{Ind_U, \upsilon\}$, which Ind_U is subset of elliptic curve group with 1024 bits and $\upsilon = xS_{NM} + k$ with 160 bits. Therefore, the user stores (1024 + 160 = 1184) bits. Also, the application server AS stores its secret key S_{AS}, which is 160 bits. Therefore, the total storage cost of ECCPWS equals to 1344 bits.

As can be seen in Table 7, the ECCPWS has the least storage cost compared to other similar ECC-based authentication protocols. In other words, on the user side, the ECCPWS, Rostampour *et al.*'s and Kumar *et al.*'s schemes have the least storage cost and also, on the server side, the ECCPWS and Sowjanya *et al.*'s protocol have the least storage cost.

8.2. Communication cost comparison

As can be seen in Table 10, in the Sowjanya *et al.*'s scheme [20], the user transmits the authentication request $\{E_U, R\}$ to the application server, which R is the subset of elliptic curve group with 1024 bits, $E_U = E_{K_{UA}}(c, U_U, right)$ where $U_U = kPK_U$ is the subset of elliptic curve group with 1024 bits, parameter $right$ is 64 bits and $c \in Z_q^*$ is 160 bits. Therefore, the authentication request length is 2272 bits. Also, the application server AS sends the authentication response $\{Auth, Y\}$ to the user, which equals to (160 + 1024 = 1184) bits. Therefore, the total communication cost of Sowjanya *et al.*'s scheme equals to 3456 bits.

In the Rostampour *et al.*'s protocol [48], the user sends the authentication message of $\{PPID_i, P_1, P_2, V_i\}$ to the server, which each parameter is 1024 bits. Therefore, the length of authentication request equals to 4096 bits. Moreover, the response of server is $\{P_3, P_4\}$ with (1024 +1024 = 2048) bits. Hence, the communication cost of Rostampour *et al.*'s protocol is 6144 bits.

In the Kumar *et al.*'s scheme [47], the user sends message $\{CID_i, X_i, Y_i, C\}$ to the server. Therefore, communication cost at the user side is (160 + 1024 + 160 + 160 = 1504) bits. Also, the communication cost of the server side consist of (1024 + 160 + 160 = 1344) bits. Since the

Table 7
The comparison of ECCPWS with other similar recent protocols in the term of storage cost (bits).

Memory capacity (bits)	[34]	[47]	[48]	[20]	ECCPWS
User	4320	1184	1184	1248	1184
Server	320	641	352	160	160
Total cost	4640	1825	1536	1408	1344

server sends massage $\{X_j, Y_j, T\}$ to the user. As a result, the total communication cost equals to 2848 bits.

In the Li *et al.*'s protocol [34], the user sends the message $M_1 = \{E_U, R, c, t_c\}$ to the application server, which its length is $((1024 + 1024 + (1024 + 1024 + 64)) + 1024 + 160 + 32 = 5376)$ bits. Moreover, the application server responses to the user with $M_2 = MAK_{SK}(y')$. The length of this message equals to 160 bits. Therefore, the total communication cost of Li *et al.*'s protocol is 5536 bits.

In the ECCPWS, the user sends the message $\{E_U, R, T_1\}$ to the application server, which R is subset of elliptic curve group elements with 1024 bits, T_1 with 32 bits and $E_U = E_{K_{UA}}(c, U_U, Ind_U, T_1)$, where $U_U = kPK_U$ and Ind_U are points in elliptic curve group with 1024 bits and $c \in {Z_q}^*$ is 160 bits. So, this request's length is 3296 bits. Also, the application server AS sends the authentication response $\{Auth, Y, T_3\}$ to the user, which its length is $(160 + 1024 + 32 = 1216)$ bits. Therefore, the total communication cost of ECCPWS equals to 4512 bits.

It can be seen that the communication cost of ECCPWS has reduced compared to ones of the Li *et al.*'s and Rostampour *et al.*'s protocols. But, it is more than the Kumar *et al.*'s and Sowjanya *et al.*'s schemes, since it provides more security and this cost must be paid for providing security. Also, the ECCPWS is more efficient in terms of the number of exchanged messages such as the Sowjanya *et al.*'s and Li *et al.*'s protocols.

8.3. Computational cost comparison

Here, we compare the ECCPWS with recent ECC-based protocols based on execution times represented in Table 9. In this table, T_h, T_s and T_m denote required times for executing hash function, encryption/decryption and ECC-based scalar point multiplication, respectively.

Since, the computational cost of the user and server are $(7T_m + 3T_h + 1T_s)$ and $(6T_m + 3T_h + 1T_s)$, respectively, so the computational cost of Sowjanya *et al.*'s scheme [20] equals to $(13T_m + 6T_h + 2T_s)$.

In the Rostampour *et al.*'s protocol [48], the computational cost of the user and server equal to $7T_m$ and $7T_m$, respectively. Therefore, the total computational cost of Rostampour *et al.*'s protocol [48] equals to $14T_m$.

In the Kumar *et al.*'s scheme [47], the computational cost of the user and server are $(6T_m + 4T_h)$ and $(8T_m + 10T_h)$, respectively. Therefore, its total computational cost equals to $(14T_m + 14T_h)$.

The computational cost in the user and server side of Li *et al.*'s protocol [34] are $(4T_m + 2T_h + 1T_s)$ and $(5T_m + 2T_h + 3T_s)$, respectively. So, the total computational cost of Li *et al.*'s protocol equals to $(9T_m + 4T_h + 4T_s)$.

The total computational cost of ECCPWS equals to $(13T_m + 6T_h + 2T_s)$. This is why the computational cost of the user and server are $(7T_m + 3T_h + 1T_s)$ and $(6T_m + 3T_h + 1T_s)$ in the ECCPWS, respectively. The results of this comparison are summarized in Table 11.

It can be seen that the computational cost of ECCPWS has reduced compared to ones of the Rostampour *et al.*'s protocol. But, it is more than ones of the Kumar *et al.*'s and Li *et al.*'s protocols. Also, the computational cost of Sowjanya *et al.*'s scheme and ECCPWS are same. Of course, it can be seen that the ECCPWS has more optimal level in the term of computational cost on the server side rather the Rostampour *et al.*'s and Kumar *et al.*'s protocols.

As a general result of comparing the performance of recent ECC-based authentication protocols, it can be inferred that the ECCPWS has reasonable storage, communication and computational costs compared to others. Since the ECCPWS is an improved version of the Sowjanya *et al.*'s protocol compared to the Sowjanya *et al.*'s scheme, one can say the storage cost has been decreased, while the computational cost is same with the Sowjanya *et al.*'s scheme. Also, the communication cost of ECCPWS is more than ones of the Sowjanya *et al.*'s scheme. This is why the ECCPWS provides more security against different insider and outsider passive and active attacks. The comparison of ECCPWS and recent similar protocols in the terms of storage cost, communication cost and computational cost are shown in Fig. 10, Fig. 11 and Fig. 12, respectively.

Table 8
The length of protocol's parameters used for performance comparision [20].

Parameter	Bit length
The elements in elliptic curve group	1024
Timestamp	32
$Right$	64
Large prime number p	512
Large prime number q	160
Random numbers	160
Secret keys	160

Table 9
The length of protocol's parameters used for performance comparision [20].

Operation	User (ms)	Server (ms)
T_h	0.0074	0.0023
T_s	0.0184	0.0046
T_m	30.67	6.38

Table 10
The communication cost comparison of ECCPWS with recent similar protocols (bits).

Communication cost (bits)	[34]	[47]	[48]	[20]	ECCPWS
User	5376	1504	4096	2272	3296
Application server	160	1344	2048	1184	1216
Total cost	5536	2848	6144	3456	4512
The number of exchanged messages in authentication phase	2	3	4	2	2

Table 11
The comparison of ECCPWS with recent similar protocols in terms of computational cost(ms).

Protocol	User (ms)	Server (ms)	Total cost (ms)
Li *et al.* [34]	$4T_m + 2T_h + 1T_s \approx 122.7132$	$5T_m + 2T_h + 3T_s \approx 31.9184$	≈ 154.6316
Kumar *et al.* [47]	$6T_m + 4T_h \approx 184.0496$	$8T_m + 10T_h \approx 51.063$	≈ 235.1126
Rostampour *et al.* [48]	$7T_m \approx 214.69$	$7T_m \approx 44.66$	≈ 259.35
Sowjanya *et al.* [20]	$7T_m + 3T_h + 1T_s \approx 214.7306$	$6T_m + 3T_h + 1T_s \approx 38.2915$	≈ 253.0221
ECCPWS	$7T_m + 3T_h + 1T_s \approx 214.7306$	$6T_m + 3T_h + 1T_s \approx 38.2915$	≈ 253.0221

9. Conclusion

Wireless Body Area Network (WBAN) has developed using medical servers and wearable health monitoring sensors, which can be used in WHMS. The collected data of patients in such systems are transmitted through insecure wireless channels. So, security and privacy of such systems are very important. Nowadays, design of secure lightweight authentication protocols for WHMS is a major challenge. In this paper, we examined the security of one ECC-based anonymous authentication scheme proposed by Sowjanya *et al.*. We proved that this scheme is vulnerable to passive insider secret disclosure and replay attacks. The complexity of these attacks is only one run of the protocol and their success probability equals to one. The vulnerability of Sowjanya *et al.*'s protocol against passive insider secret disclosure attack has lead that any other attack is applicable to it, such as the NM impersonation and traceability attacks and etc.. We also proposed an improved authentication scheme based on ECC called ECCPWS. The security analysis of ECCPWS, which performed informally and formally through ROR model, BAN logic and automatic security verification tools such as Scyther and AVISPA tools, show the ECCPWS overcomes to all security loopholes of its predecessor with reasonable computational, communication and storage costs compared to related schemes.

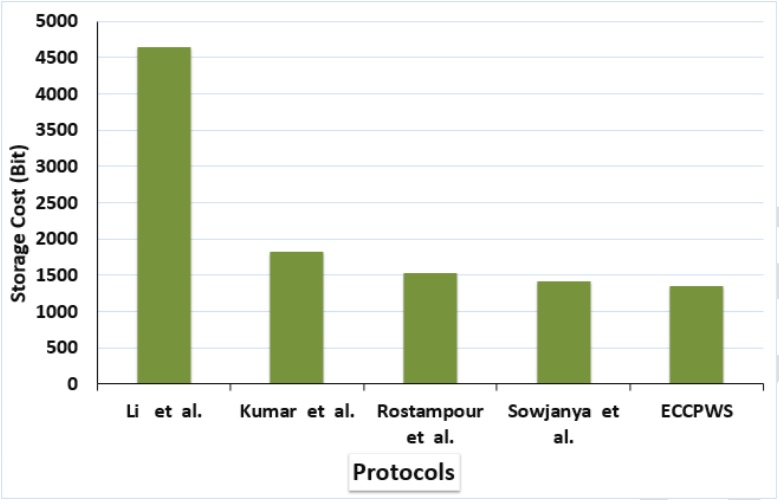

Figure 10: The storage cost comparison of ECCPWS with similar related protocols.

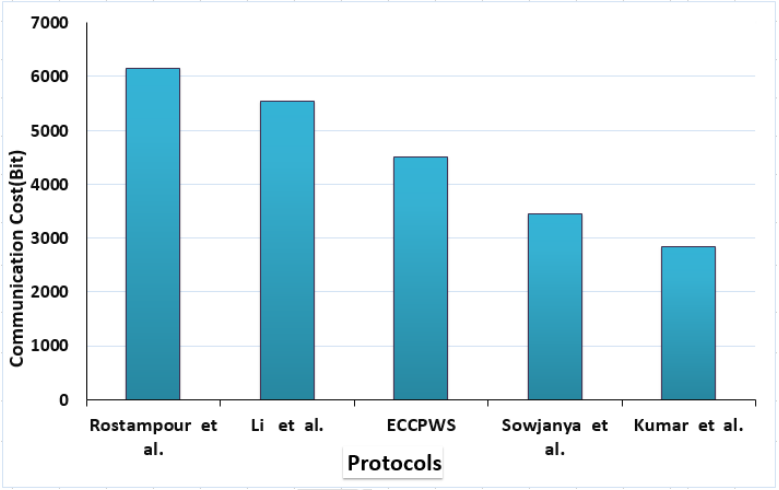

Figure 11: The communication cost comparison of ECCPWS with similar related protocols.

CRediT authorship contribution statement

Fatemeh Pirmoradian: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Software, Writing - original draft, Writing - review & editing. **Masoumeh Safkhani:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Validation, Supervision, Formal analysis, Writing - original draft, Writing - review & editing. **Seyed Mohammad Dakhilalian:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Validation, Supervision, Formal analysis, Writing - original draft, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments, which helped us improve the manuscript. This work was supported by Isfahan University of Technology (IUT).

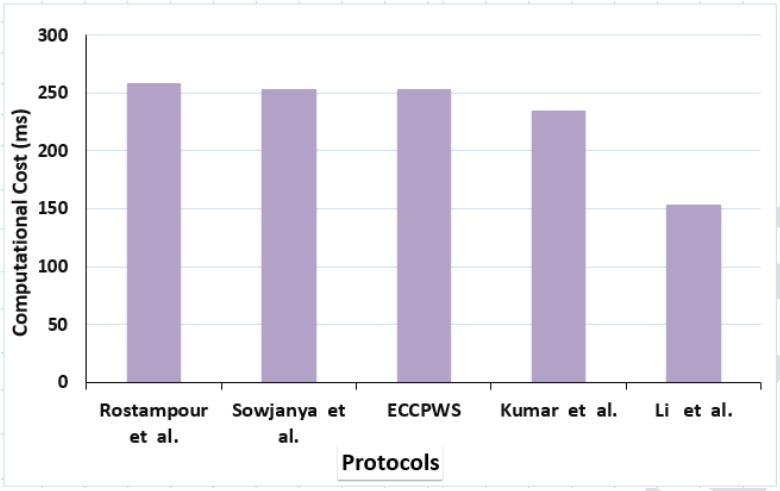

Figure 12: The computational cost comparison of ECCPWS with similar related protocols.

List of Acronyms

Adv Advantage
AS Application Server
AVISPA Automated Validation of Internet Security Protocols and Application
BAN Burrows-Abadi-Needham
CL-AtSe Constraint Logic based Attack Searcher
DL Discrete Logarithm
ECDDHP Elliptic Curve Decisional Diffie-Hellman Problem
ECCPWS ECC-based Protocol for WBAN Systems
ECDLP Elliptic Curve Discrete Logarithm Problem
ECC Elliptic Curve Cryptography
ECCDHP Elliptic Curve Computational Diffie-Helman Problem
GNY Gong Needham Yahalom
HLPSL High Level Protocols Specification Language
IF Intermediate Format
IR Inference Rules
ID Identification
IoT Internet of Things
NM Network Manager
OF Output Format
OFMC On the Fly Model Checker
ROR Real-Or-Random
RSA Rivest Shamir Adleman
SPAN Security Protocol ANimator for AVISPA
SATMC SAT Based Model Checker
SVO Syverson-Van- Oorschot
SPDL Security Protocol Description Language
SK Session Key
TMIS Telecare Medicine Information Systems
TA4SP Tree Automata based Protocol Analyzer
U User
VO Van Oorschot
WBAN Wireless Body Area Networks
WHMS Wearable Health Monitoring Systems
WSN Wireless System Network

References

[1] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C.M.Leung. Body Area Networks: A Survey. *Mobile Networks and Applications.*, 16:171–193, 2011.

[2] M. Safkhani, C. Camara, P. Peris-Lopez, and N. Bagheri. RSEAP2: An enhanced version of RSEAP, an RFID based authentication protocol for vehicular cloud computing. *Vehicular Communications*, 28, 2021.

[3] A. Gupta, M. Tripathi, and A. Sharma. A provably secure and efficient anonymous mutual authentication and key agreement protocol for wearable devices in WBAN. *Computer Communications*, 160:311–325, 2020.

[4] H. Arshad, V. Teymoori, M. Nikooghadam, and H. Abbassi. On the security of a two-factor authentication and key agreement scheme for telecare medicine information systems. *Journal of Medical Systems*, 39(76), 2015.

[5] J. Mo, W. Shen, and W. Pan. An improved anonymous authentication protocol for wearable health monitoring systems. *Wireless Communications and Mobile Computing*, 2020.

[6] X. Li, J. Peng, M. Obaidat, S.Obaidat, F. Wu, M. Khurram Khan, and C. Chen. A Secure Three-Factor User Authentication Protocol With Forward Secrecy for Wireless Medical Sensor Network Systems. *IEEE Systems Journal*, 14(19428538):39–50, 2019.

[7] R. Amin, S.K. Hafizul Islam, G.P. Biswas, M. Khurram Khan, and N. Kumar. A robust and anonymous patient monitoring system using wireless medical sensor networks. *Future Generation Computer Systems*, 80:483–495, 2018.

[8] S.A. Chaudhry, M. Tawab Khan, M. khurram Khan, and T. Shon. A Multiserver Biometric Authentication Scheme for TMIS using

Elliptic Curve Cryptography. *Journal of Medical Systems*, 40(230), 2016.

[9] N. Yessad, S. Bouchelaghem, F.S. Ouada, and M. Omar. Secure and reliable patient body motion based authentication approach for medical body area networks. *Parvasive and Mobile Computing*, 42:351–370, 2017.

[10] M. Yavari, M. Safkhani, S. Kumari, and C.M. Chen. An Improved Blockchain-Based Authentication Protocol for IoT Network Management. *Security and Communication Networks*, 2020(8836214), 2020.

[11] D. He, S. Zeadally, N. Kumar, and J.H. Lee. Anonymous Authentication for Wireless Body Area Networks With Provable Security. *IEEE Systems Journal*, 11(17390228):2590–2601, 2016.

[12] H. Debiao, C. Jianhua, and Z. Rui. A More Secure uthentication scheme for Telecare Medicine Information systems. *Journal of Medical Systems*, 36:1989–1995, 2012.

[13] X. Yan, W. Li, P. Li, J. Wang, and P. Gong. A Secure Biometrics-based Authentication Scheme for Telecare Medicine Information Systems. *Journal of medical systems*, 37(9972), 2013.

[14] Z. Zhu. An Efficient Authentication Scheme for Telecare Medicine Information Systems. *Journal of Medical Systems*, 36:3833–3838, 2012.

[15] S.A. Chaudhry, H. Naqvi, K. Mahmood, H.F. Ahmad, and M. Khurram Khan. An Improved Remote User Authentication Scheme Using Elliptic Curve Cryptography. *Wireless Personal Communications*, 96:5355–5373, 2017.

[16] C.T Li, D.H. Shih, and C.C Wang. Cloud-assisted mutual authentication and privacy preservation protocol for telecare medical information systems. *Computers Methods and Programs in Biomedicine*, 157:191–203, 2018.

[17] Z. Tan. A User Anonymity Preserving Three-Factor Authentication Scheme for Telecare Medicine Information Systems. *Journal of Medical Systems*, 38(16), 2014.

[18] M. Safkhani, S. Rostampour, Y. Bendavid, and N. Bagheri. IoT in medical and pharmaceutical: Designing lightweight RFID security protocols for ensuring supply chain integrity. *Computer Networks*, 181, 2020.

[19] H. Arshad and M. Nikooghadam. Three-Factor Anonymous Authentication and Key Agreement Scheme for Telecare Medicine Information systems. *Journal of Medical Systems*, 38(136), 2014.

[20] K. Sowjanya, M. Dasgupta, and S. Ray. An elliptic curve cryptography based enhanced anonymous authentication protocol for wearable health monitoring systems. *International Journal of Information Security*, 19:129–146, 2020.

[21] L. Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, 1981.

[22] M. Abdul Azim and A. Jamalipour. An efficient elliptic curve cryptography based authenticated key agreement protocol for wireless LAN security. *HPSR. 2005 Workshop on High Performance Switching and Routing*, 2005.

[23] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov. System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of Mobile Multimedia*, 1(4):307–326, 2006.

[24] D. Boyle and T. Newe. A Survey of Authentication Mechanisms: Authentication for Ad-Hoc Wireless Sensor' networks. *IEEE Sensors Applications Symposium*, 2007.

[25] I.B. Preneel and D. Singelee. Study and design of a security architecture for wireless personal area networks. 2008.

[26] X.H Le, M. Khalid, R. Sankar, and S. Lee. An Efficient Mutual Authentication and Access Control Scheme for Wireless Sensor Networks in Healthcare. *Journal of Networks*, 6(3), 2011.

[27] J. Liu and K.S. Kwak. Hybrid security mechanisms for wireless body area networks. *2010 Second International Conference on Ubiquitous and Future Networks (ICUFN)*, 71(11475046), 2010.

[28] M. Mana, M. Feham, and B.A. Bensaber. Trust Key Management Scheme for Wireless Body Area Networks. *International Journal of Network Security*, 12(2):75–83, 2011.

[29] C.K. Yeh, H.M. Chen, and J.W. Lo. An Authentication Protocol for Ubiquitous Health Monitoring Systems. *Journal of Medical and Biological Engineering*, pages 415–419, 2013.

[30] Z. Zhao. An Efficient Anonymous Authentication Scheme for Wireless Body Area Networks Using Elliptic Curve Cryptosystem. *Journal of Medical systems*, 38(13), 2014.

[31] D. He and S. Zeadally. Authentication protocol for an ambient assisted living system. *IEEE Communications Magazine*, 53(14863802):71–77, 2015.

[32] F. Wu, X. Li, Lili. Xu, S. Kumari, M. Karuppiah, and J. Shen. A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server. *Computer and Electrical Engineering*, 63(14863802):168–181, 2017.

[33] J. Liu, L. Zhang, and R. Sun. 1- RAAP: An Efficient 1-Round Anonymous Authentication Protocol for Wireless Body Area Networks. *wireless communications*, 16(5), 2016.

[34] X. Li, J. Peng, S. Kumari, F. Wu, M. Karuppiah, K. Kwang, and R. Choo. An enhanced 1-round authentication protocol for wireless body area networks with user anonymity. *Computers and Electrical Engineering*, 61:238–249, 2017.

[35] S.K. Hafizul Islam, R. Amin, G.P. Biswas, M. Sabzinejad Farash, X. Li, and S. Kumari. An improved three party authenticated key exchange protocol using hash function and elliptic curve cryptography for mobile-commerce environments. *Journal of King Saud University-Computer and Information Science*, 29(3):311–324, 2017.

[36] M. Safkhani, N. Bagheri, S. Kumari, H.R Tavakoli, S. Kumar, and J. Chen. RESEAP: an ECC-based authentication and key agreement scheme for IoT applications. *IEEE Access*, 8(20140216):200851–200862, 2020.

[37] J. Hoffstein, J. Pipher, and J.H Silverman. An Introduction to Mathematical Cryptography. 2008.

[38] A. Kumari, S. Jangirala, M.Y. Abbasi, V. Kumar, and M. Alam. ESEAP: ECC based secure and efficient mutual authentication protocol using smart card. *Journal of Information Security and Applications*, 51(102443), 2020.

[39] A.K. Das, M. Wazid, A.R. Yannam, J.J.P.C Rodrigues, and Y. Park. Provably Secure ECC-based Device Access Control and Key Agreement Protocol for IoT Environment. *IEEE Access*, 7(18648442):55382–55397, 2019.

[40] A.K. Das, M. Wazid, N. Kumar, A.V Vasilakos, and J.J.P.C Rodrigues. Biometrics-Based Privacy-Preserving User Authentication Scheme for Cloud-Based Industrial Internet of Things Deployment. *IEEE Internet of Things Journal*, 5(6):4900–4913, 2018.

[41] J. Qi, M. Jianfeng, Y. Chao, M. Xindi, S. Jian, and S.A. Chaudhry. Efficient end-to-end authentication protocol for wearable health monitoring systems. *Computers and Electrical Engineering*, 63:182–195, 2017.

[42] J. Wei, X. Hu, and W. Liu. An Improved Authentication Scheme for Telecare Medicine Information Systems. *Journal of Medical Systems*, 36:3597–3604, 2012.

[43] A.K. Das, S. Zeadally, and M. Wazid. Lightweight authentication protocols for wearable devices. *Computers and Electrical Engineering*, 63:196–208, 2017.

[44] M. Safkhani and N. Bagheri. Passive secret disclosure attack on an ultralightweight authentication protocol for internet of things. *The Journal of Supercomputing*, 73:3579–3585, 2017.

[45] M. Safkhani and M. Shariat. Implementation of secret disclosure

attack against two IoT lightweight authentication protocols. *The Journal of Supercomputing*, 74:6220–6235, 2018.

[46] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29:198–208, 1983.

[47] A. Kumar, K. Abhishek, X. Liu, and A. Holdorai. An Efficient Privacy-Preserving ID Centric Authentication in IoT Based Cloud Servers for Sustainable Smart Cities. *Wireless Personal Communications*, 117:3229–3253, 2021.

[48] S. Rostampour, M. Safkhani, Y. Bendavid, and N. Bagheri. ECCbAP: A secure ECC-based authentication protocol for IoT edge devices. *Pervasive and Mobile Computing*, 67(101194), 2020.

[49] C.J.F Cremers. The Scyther Tool: Verification, Falsification, and Analysis of Security protocols. *International Conference on Computer Aided Verification*, pages 414–418, 2008.

[50] M. Burrows, M. Abadi, and R.M. Needham. A logic of authentication. *Proceedings Mathematical Physical and Engineering Sciences*, 426:233–271, 1989.

[51] L. Takkinen. Analysing Security Protocols with AVISPA. *TKK T-110.7290 Research Seminar on Network Security*, 12(1), 2006.

[52] L. Gong, R. Needham, and R. Yahalom. Reasoning about Belief in Cryptographic Protocols. *IEEE Symposium on Security and Privacy*, pages 234–248, 1990.

[53] R. Kusters and T. Truderung. Using ProVerif to Analyze Protocols with Diffie-Hellman Exponentiation. *2009 22nd IEEE Computer Security Foundations Symposium*, 2009.

Appendix A: Pseudo code of ECCPWS in the Scyther tool

```
hashfunction h;
hashfunction ECC;
const ADD: Function;
secret B;
secret idu;
secret t;
secret y;
secret r;
usertype Timestamp;
macro   IndU=ECC(t,B);
macro   UU= {t}pk(U);
macro   x=h(UU,IndU);
macro   v=ADD({x}sk(NM),t);
macro   R=ECC(r,B);
macro   Qprim={R}sk(AS);
macro   c=ADD(v,r, sk(U));
macro   EU={C,UU,IndU,T1}pk(AS);
macro   Y=ECC(y,B);
macro   Auth=h(UU,R, Qprim,Y,T3);
macro   M=ECC(idu,B);
protocol @oracle (X){
role X {
var Y:Agent;
const P;
recv_!X1(X, X, ECC(X,ECC(Y,P)));
send_!X2(X, X, ECC(Y,ECC(X,P)));
}
}
protocol @mad (X){
role X {
var Y:Agent;
const P;
recv_!X1(X,X,ECC(sk(Y),pk(X)));
send_!X2(X,X,ECC(sk(X),pk(Y)));
}
}
protocol @oracleM (X){
role X {
var Y:Agent;
const P;
var  r;
fresh y;
recv_!X1(X,X,ECC(r,ECC(y,B)));
send_!X2(X,X,ECC(y,ECC(r,B)));
}
}
protocol proposed (U,NM, AS){
role U {
secret  B;
secret  idu;
fresh  r;
fresh  T1: Timestamp;
fresh  T4: Timestamp;
var  y;
var  t;
var  T3: Timestamp;
var  T2: Timestamp;
send_1 (U, NM, {M,PK(U)}k(U,NM));
recv_2 (NM, U, {IndU,v} k(U,NM));
send_3 (U, AS, EU,R,T1));
recv_4 (AS, U, Auth,Y,T3);
claim(U, Alive);
claim(U, Weakagree);
claim(U, Niagree);
claim(U, Secret, sk(U));
claim(U, Secret,r);
};
role  NM {
secret  B;
secret  idu;
fresh  t;
recv_1 (U, NM, {M,PK(U)}k(U,NM));
send_2 (NM, U, {IndU,v}k(U,NM));
claim(NM, Alive);
claim(NM, Weakagree);
claim(NM, Niagree);
claim(NM, Secret, sk(NM));
};
role AS {
var r;
fresh y;
fresh T2: Timestamp;
fresh T3: Timestamp;
var t;
var T1: Timestamp;
var T4: Timestamp;
secret B;
secret idu;
recv_3 (U, AS, EU,R,T1);
recv_4 (AS, U, Auth,Y,T3);
claim(AS, Alive);
claim(AS, Weakagree);
claim(AS, Niagree);
claim(AS, Secret, sk(AS));
};
};
```

Appendix B: Pseudo code of ECCPWS in the AVISPA tool

```
role alice(U, S, A:agent, SKus:
symmetric_key, H,Add:hash_{func},
SND,RCV:channel(dy))
played_by U
def=local State:nat,
SU, B, PKU, IDU, UU, X, PKas, T1, T3,
R,Q, kua, C, AuthU, SK:text, ID, IndU,
V, RR, EU, Auth, YY:text,
F:hash_func
const auth_1, auth_2, auth_3,
subs1, subs2, subs3, subs4, subs5, subs6:
protocl_id
init State:=0
transition
1. State=0 /\ RCV(start) =|>
State ':= 1 /\ SU': new() /\ IDU':=new()
/\ PKU':=F(SU',B) /\ ID':=F(IDU',B)
/\ secret ({SU,IDU}, subs1, {U})
```

```
/\ SND ({ID'}_SKus . {PKU'}_SKus)
2. State=1 /\RCV({IndU'}_SKus .{V'}\_SKus)=|>
State ':= 2 /\ UU':= F(SU, IndU)
/\ X':=H(UU' . IndU) /\ kua':=F(SU, PKas)
/\ R':=new() /\ T1':=new()
/\ RR':=F(R', B) /\  Q':=F(R' , PKas)
/\ C':=Add(V . R' . SU)
/\ EU':={C' . UU'. IndU . T1'}_SKus
/\ SND (EU' . RR' . T1)
/\ secret ({R' . C' . UU'}, subs2, {U,A})
/\ witness(U, A, auth_1, C)
/\ request(S, U, auth_3, UU)
3. State=2 /\ RCV(Auth' . YY' . T3') =|>
State ':= 3
/\  AuthU':= H(UU .  RR . Q . YY' . T3')
/\ SK':=H(AuthU' . UU . R,YY')
/\ secret ({SK'}, subs3, {U,A})
/\ request(A, U, auth_2, Auth')
end role
role server (S, U, A:agent, SKus:
symmetric_key,H,Add:hash_func,
SND,RCV:channel(dy))
played_by S
def=
local State:nat,
SN, UU, K, X, B, PKU:text,
ID, IndU, V:text,
F:hash_func
const auth_1, auth_2, auth_3,
subs1,subs2,subs3,subs4,
subs5,subs6:protocol_id
init State:=0
transition
1. State=0 /\ RCV({ID}SKus . {PKU}_SKus) =|>
State ':= 1 /\ SN': new()
/\ K':=new() /\ IndU':=F(K' , B)
/\ UU':=F(K', PKU) /\ X':=H(UU' . IndU)
/\ V':=Add(F(X', SN) . K')
/\ SND ({IndU}_SKus . {V'}_SKus)
/\ secret ({SN'}, subs6, {S})
/\ witness(S, U, auth_3, UU)
end role
role bob (A, S, U: agent, SKus:
symmetric_key, H,Add:hash_func,
SND,RCV:channel(dy))
played_by A
def=
local State:nat,
SA, SU, PKas, B, Q, X, PKU, KKua, T1,
T3, Z, Y, UU, C:text, YY, Auth, EU,
RR, IndU: text,
F:hash_func
const auth_1, auth_2, auth_3,
subs1, subs2, subs3, subs4, subs5,
subs6: protocol_id
init State:=0
transition
1. State=0 /\ RCV(EU' . RR . T1') =|>
State ':= 1 /\ SA': new()
/\  PKas':=F(SA' , B)
/\ secret ({SA'}, subs5, {U, A, S})
/\ QQ':=F(SA , RR') /\ KKua':=F(SA, PKU')
/\ X':=H(UU' . IndU) /\ Y':=new()
/\ T3':=new() /\ YY':=F(Y', B)
/\ Z':=F(Y' , RR')
/\ Auth':=H(UU . RR' . QQ . YY' . T3')
/\ SK':=F(Auth' . UU . Z')
/\ SND (Auth' . YY' . T3')
/\ secret ({SK}, subs4, {A,U})
/\ request(U, A, auth_1, C)
/\ witness(S, U, auth_2, Auth')
end role
role session (U, S, A: agent, SKus:
symmetric_key, Add:hash_func)
def=local
SND1, RCV1, SND2, RCV2, SND3, RCV3:
channel(dy)
/\composition
alice(U, S, A, SKus, H, Add, SND1, RCV1)
/\server(U, S, A, SKus, H, Add, SND2, RCV2)
/\bob(U, S, A, SKus, H, Add, SND3, RCV3)
end role
role environment()
def=const u,s,a:agent,
skus:symmetric_key,
f,h,add: hash_func,
v,eu,rr,auth,yy,indu,id:text,
auth_1, auth_2, auth_3, subs1, subs2,
subs3, subs4, subs5, subs6: protocl_id
intrusder_knowledge = {u, s, a, h, f, add, id,
indu, v, eu, rr, auth, yy}
composition
session(u, s, a, skus, h, add)
/\session(u, s, a, skus, h, add)
/\session(u, s, a, skus, h, add)
end role
goal
secrecy_of subs1
secrecy_of subs2
secrecy_of subs3
secrecy_of subs4
secrecy_of subs5
secrecy_of subs6
%authentication_on auth_1
%authentication_on auth_2
%authentication_on auth_3
end goal
environment()
```

Fatemeh Pirmoradian is currently a PHD candidate at Electrical and Computer Engineering Department, Isfahan University of Technology (IUT), Isfahan, Iran. She received her M.Sc. and B.Sc. degrees in Electrical Engineering department, Shahid Rajaee Teacher Training University in 2016 and University of Isfahan in 2012, respectively. She is interested in design and security analysis of ECC based cryptography authentication protocols used in TMIS systems.

Masoumeh Safkhani is an Associate Professor at Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. She received her Ph.D. in Electrical Engineering from Iran University of Science and Technology (IUST), 2012, with the security analysis of RFID protocols as her major field. Her current research interests include the security analysis of lightweight and ultra-lightweight protocols, targeting constrained environments such as RFID, IoT, VANET and WSN. She is the author/coauthor of more than 50 technical articles in information security and cryptology in major international journals and conferences.

Seyed Mohammad Dakhilalian received the B.Sc. and Ph.D. degrees in Electrical Engineering from Isfahan University of Technology (IUT) in 1989 and 1998, respectively and M.Sc. degree in Electrical Engineering from Tarbiat Modarres University in 1993. He was an Assistant Professor of Faculty of Information and Communication Technology, Ministry of ICT, Tehran, Iran in 1999-2001. He joined IUT in 2001 and at present time is an Associate Professor in Electrical and Computer Engineering Department. His current research interests are Cryptography and Data Security.

CRediT authorship contribution statement

Fatemeh Pirmoradian: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Software, Writing - original draft, Writing - review & editing.

Masoumeh Safkhani: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Validation, Supervision, Formal analysis, Writing - original draft, Writing - review & editing.

Seyed Mohammad Dakhilalian: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Validation, Supervision, Formal analysis, Writing - original draft, Writing - review & editing.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: