J Med Syst (2016) 40: 237
DOI 10.1007/s10916-016-0585-3

SYSTEMS-LEVEL QUALITY IMPROVEMENT

Design of a Secure Authentication and Key Agreement Scheme Preserving User Privacy Usable in Telecare Medicine Information Systems

Hamed Arshad[1] · **Abbas Rasoolzadegan**[1]

Received: 4 March 2016 / Accepted: 4 September 2016 / Published online: 24 September 2016

Abstract Authentication and key agreement schemes play a very important role in enhancing the level of security of telecare medicine information systems (TMISs). Recently, Amin and Biswas demonstrated that the authentication scheme proposed by Giri et al. is vulnerable to off-line password guessing attacks and privileged insider attacks and also does not provide user anonymity. They also proposed an improved authentication scheme, claiming that it resists various security attacks. However, this paper demonstrates that Amin and Biswas's scheme is defenseless against off-line password guessing attacks and replay attacks and also does not provide perfect forward secrecy. This paper also shows that Giri et al.'s scheme not only suffers from the weaknesses pointed out by Amin and Biswas, but it also is vulnerable to replay attacks and does not provide perfect forward secrecy. Moreover, this paper proposes a novel authentication and key agreement scheme to overcome the mentioned weaknesses. Security and performance analyses show that the proposed scheme not only overcomes the mentioned security weaknesses, but also is more efficient than the previous schemes.

Keywords Authentication · Key agreement · Telecare medicine information systems · Security

This article is part of the Topical Collection on *Systems-Level Quality Improvement*

✉ Abbas Rasoolzadegan
rasoolzadegan@um.ac.ir

✉ Hamed Arshad
hamedarshad@aol.com; hamedarshad@stu.um.ac.ir

[1] Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

Introduction

Growth of the aging population causes an increase in the rate of chronic diseases such as diabetes, cardiovascular diseases, and mental illnesses. Such diseases require long-term treatment with the frequent hospital/clinic-based checkups, which in turn induces excessive costs and stress on the patients (due to the repeated trips to the hospital). This causes significant adverse effects on the patient's quality of life [1]. Without regular monitoring and medical care, chronic diseases can cause critical conditions for the patients. Therefore, developing a system that can enable patients diagnosed with chronic diseases to receive remote treatment at home is useful for both the patients and the medical infrastructure (facilities, doctors, staff, etc.) [2]. In fact, providing home-based long-term medical care services for chronic patients enhances the quality of their lives.

Nowadays, information and communication technologies are increasingly used in the medical sector to improve and facilitate healthcare delivery services. For example, telecare medicine information systems (TMISs) enable patients and doctors to access medical services and information at anytime and anywhere via the Internet [3–5]. By employing TMIS, patients without leaving home can obtain the same medical services as at hospital. Specifically, patients in rural areas are no longer required to travel long distances to visit a doctor. The medical staffs can remotely monitor the health condition of the patients and physicians can treat patients in a remote place at the right time and lower cost. Therefore, TMISs provide more convenience for patients and reduce the patients' expenses such as travel and hospitalization costs. Besides, the patients' medical records stored in the medical servers of TMIS allow doctors to provide more accurate diagnoses and prescribe better treatments [6].

Due to the open architecture of the Internet, TMISs that work based on the Internet are subject to various security attacks [7, 8]. As shown in Fig. 1, an adversary may capture the messages exchanged between a patient and the medical server and obtain the confidential information about the patient. It is obvious that disclosure of the health information about the patient breaches the privacy of the patient. The adversary may also modify the messages exchanged between the physician and the patient and cause irreparable injury to the patient. Hence, a secure mechanism for authentication and key agreement should be employed to restrict unauthorized accesses to the medical information stored on the medical servers and exchanged between users (physicians and patients) and medical servers [9–11]. Hitherto, numerous authentication and key agreement schemes have been proposed for TMISs. Recently, Amin and Biswas [12] analyzed the security of the authentication scheme proposed by Giri et al. [13] and presented some attacks on it. Then, they proposed an improved authentication scheme for TMISs and claimed that their improved scheme provides an acceptable level of security. However, we show that Amin and Biswas's scheme [12] is insecure against some security attacks and does not provide perfect forward secrecy. We also demonstrate that Giri et al.'s scheme [13] not only suffers from the weaknesses identified by Amin and Biswas, but it also is vulnerable to replay attacks and does not provide perfect forward secrecy. Furthermore, in order to improve the security and efficiency of the previous schemes, we propose a new authentication and key agreement scheme using the elliptic curve cryptosystem (ECC).

The rest of the paper is organized as follows. Related works are listed in "Related works". "Review of Giri et al.'s scheme" briefly reviews Giri et al.'s scheme. "Weaknesses of Giri et al.'s scheme" presents the weaknesses of Giri et al.'s scheme. In "Review of Amin and Biswas's scheme", Amin and Biswas's scheme is reviewed. In "Weaknesses of Amin and Biswas's scheme", weaknesses of Amin and Biswas's scheme are discussed. In "The proposed scheme", the proposed scheme is described. "Security analysis" and "Performance analysis" analyze the security and performance of the proposed scheme. Finally, a conclusion is given in "Conclusion".

Related works

Until now, a large number of authentication and key agreement schemes have been proposed. However, most of them have been proved to be insecure against various security attacks.

In 1981, Lamport [14] proposed the first authentication scheme using one-way hash functions. Since Lamport's scheme does not need time-consuming cryptographic operations, it is a lightweight authentication scheme. However, Lennon et al. [15] and Yen and Liao [16] demonstrated that Lamport's scheme is vulnerable to stolen verifier attacks. The vulnerability of Lamport's scheme lies in the fact that in the scheme, the server maintains the hashed values of the users' passwords. Lamport's scheme falls in the category of one-factor authentication schemes, because the

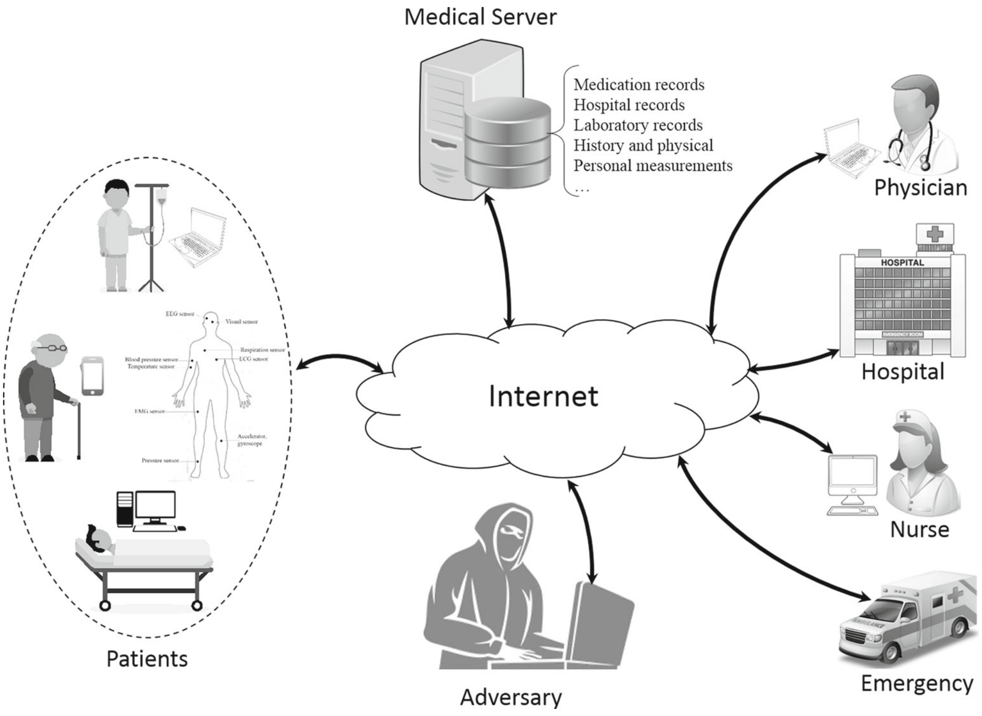


Fig. 1 An overall scheme of the application of TMIS

server authenticates the users just through their passwords. Typically, in one-factor authentication schemes, the server maintains a table containing the verifiers of the users [17, 18]. Hence, the servers are often the favorite targets of adversaries, because if an adversary achieves the verifier of a user that is stored in the verification table, then he/she can masquerade as the victim user [19–22].

In order to overcome stolen verifier attacks and enhance the security, Hwang and Li [23] proposed another type of authentication called two-factor authentication. Typically, in two-factor authentication schemes, the server does not need to maintain the verifiers of users. Instead, the server stores some personalized information into a smart card and gives the smart card to the user at the end of the registration process. Hence, if an adversary wants to impersonate a user, he/she has to obtain both the password and smart card of the user [24, 25]. Since the scheme of Hwang and Li [23] was a two-factor authentication scheme and the security of it was based on the difficulty of solving the Discrete Logarithm Problem (DLP), Hwang and Li [23] claimed that their scheme is a secure authentication scheme. Nevertheless, Chan and Chen [26] demonstrated that Hwang and Li's scheme [23] is defenseless against impersonation attacks. Sun et al. in [27] proposed a lightweight two-factor authentication scheme, claiming that it could resist security attacks. In [28] Chien et al. demonstrated that the scheme of Sun et al. [27] does not provide an acceptable level of the security and then suggested an improved authentication scheme. Unfortunately, Ku and Chen [29] proved that the scheme suggested by Chien et al. [28] is also susceptible to insider attacks and parallel session attacks. Ku and Chen [29] also proposed an improved authentication scheme to overcome the weaknesses of Chien et al.'s scheme [28]. However, Yoon et al. [30] pointed out that Ku and Chen's scheme cannot resist parallel session attacks and denial of service attacks. In order to enhance the security, Yoon et al. [30] proposed a new authentication scheme. Nevertheless, in [31] it is demonstrated that both the schemes proposed in [29, 30] are susceptible to password guessing attacks, impersonation attacks, and denial of service attacks.

In 2012, in order to enhance the security of the previous schemes, Hsieh and Leu [32] proposed a novel authentication scheme. However, Wang et al. [33] demonstrated that Hsieh and Leu's scheme is defenseless against password guessing attacks. Then, they suggested an improved scheme with the claim that it could withstand various security attacks. Chang et al. in [34] claimed that Wang et al.'s scheme [33] does not preserve user privacy because the user uses the same identity for all the sessions. Then, Chang et al. [34] proposed an improved scheme with the claim that it withstands various attacks and preserves user privacy. However, Kumari et al. [35] pointed out that the scheme proposed by Chang et al. [34] cannot withstand password guessing attacks and impersonation attacks. Moreover, they proposed a lightweight authentication scheme, claiming that it provides an acceptable level of the security. Nevertheless, in [7] it is proved that Kumari et al.'s scheme [35] is susceptible to password guessing attacks and does not preserve user privacy.

In 2015, Giri et al. [13] proposed an improved authentication and key agreement scheme [13] and claimed that their scheme could withstand various attacks. However, Amin and Biswas [12] demonstrated that Giri et al.'s scheme is vulnerable to off-line password guessing attacks and privileged insider attacks and also does not provide user anonymity. Then, in order to overcome the weaknesses of Giri et al.'s scheme, Amin and Biswas [12] proposed an improved authentication scheme for TMISs. This paper demonstrates that Amin and Biswas's scheme [12] is vulnerable to off-line password guessing attacks and replay attacks and also does not provide perfect forward secrecy. The paper also shows that Giri et al.'s scheme [13] not only suffers from the weaknesses demonstrated by Amin and Biswas, but it also is vulnerable to replay attacks and does not provide perfect forward secrecy.

Review of Giri et al.'s scheme

This section briefly reviews Giri et al.'s authentication and key agreement scheme [13]. Giri et al.'s scheme includes five phases, i.e., initialization phase, registration phase, login phase, authentication and session key agreement phase, and password change phase. Since the password change phase of Giri et al.'s scheme is not relevant to our analysis, we only review the first four phases. The notations used in Giri et al.'s scheme are listed in Table 1.

Initialization phase

In this phase, the server chooses two large primes p and q and computes $n = p \times q$. Then, the server chooses a secure one-way hash function $h(\cdot) : \{0, 1\}^* \rightarrow Z_q^*$ and two integers

Table 1 Notations

Symbol	*Description*
PW_i	The user's password
ID_i	The user's identity
SK	The shared session key between the user and the server
e	The server's public key
d	The server's secret key
$\|$	The concatenation operation
$\oplus$	The exclusive-or (XOR) operation
$\times$	The modular multiplication operation

e and d such that $e \times d \bmod (p-1)(q-1) = 1$. Finally, the server keeps d as its secret key and publishes e as its public key.

Registration phase

In this phase, as shown in Fig. 2, a new user can register with the server and obtain a personalized smart card as follows:

Step 1. The user chooses his/her identity ID_i and password PW_i and selects a random number b_i. Then, the user computes $PWb_i = h(PW_i \parallel b_i)$ and sends a message $\{ID_i, PWb_i\}$ to the server through a secure channel.

Step 2. Upon receiving the message $\{ID_i, PWb_i\}$, the server computes $R_i = h(ID_i \parallel d)$, $B_i = (PWb_i \parallel R_i)^e \bmod n$, $A_i = R_i \oplus PWb_i$, and $L_i = h(R_i \parallel PWb_i)$, stores $\{ID_i, A_i, B_i, L_i, h(\cdot)\}$ into a smart card, and sends the smart card to the user through the secure channel.

Step 3. After receiving the smart card, the user stores the random number b_i in the memory of the smart card.

Login phase

When a user wants to login to the server, he/she inserts his/her smart card into the card reader and enters his/her identity ID_i and password PW_i. Then, the smart card computes $PWb_i = h(PW_i \parallel b_i)$ and $R_i = A_i \oplus PWb_i$ and checks whether $h(R_i \parallel PWb_i)$ is equal to the stored L_i or not. If they are not equal, the smart card halts the process. Otherwise, the smart card selects a random number N_1, computes $C_i = h(PWb_i \parallel N_1 \parallel R_i)$ and $D_i = PWb_i \oplus N_1$, and sends a message $\{ID_i, C_i, B_i, D_i\}$ to the server through a public channel.

Authentication and session key agreement phase

In this phase, as shown in Fig. 3, the user and the server verify the authenticity of each other and negotiate a session key as follows:

Step 1. Upon receiving the message $\{ID_i, C_i, B_i, D_i\}$, the server checks whether the received identity is valid or not. If it is not a valid identity, the server ignores the received message. Otherwise, the server decrypts B_i as $(B_i)^d \bmod n = (PWb_i^* \parallel R_i^*)$, computes $R_i = h(ID_i \parallel d)$, and checks whether the decrypted R_i^* is equal to the computed R_i or not. If they are not equal, the server terminates the session; otherwise, it computes $N_1^* = PWb_i^* \oplus D_i$ and checks whether $h(PWb_i^* \parallel N_1^* \parallel R_i)$ is equal to the received C_i or not. If they are not equal, the server terminates the session. Otherwise, the server authenticates the user, accepts his/her login request, selects a random number N_2, and computes $N_3 = N_1^* \oplus N_2$ and $K_i = h(R_i \parallel N_2)$. Finally, the server computes the session key $SK = h(ID_i \parallel PWb_i^* \parallel N_1^* \parallel N_2)$ and sends a message $\{N_3, K_i\}$ to the user through the public channel.

Step 2. After receiving the message $\{N_3, K_i\}$, the user computes $N_2^* = N_3 \oplus N_1$ and checks whether $h(R_i \parallel N_2^*)$ is equal to the received K_i or not. If they are not equal, the user terminates the session. Otherwise, the user authenticates the server and computes the session key SK as $SK = h(ID_i \parallel PWb_i \parallel N_1 \parallel N_2^*)$.

Weaknesses of Giri et al.'s scheme

Recently, Amin and Biswas [12] pointed out that Giri et al.'s scheme [13] is vulnerable to off-line password guessing attacks and does not provide user anonymity. This section demonstrates that Giri et al.'s scheme [13] not only suffers from the weaknesses pointed out by Amin and Biswas [12], but it also is vulnerable to replay attacks and does not support perfect forward secrecy. The details are as follows.

Replay attacks

Suppose an adversary has eavesdropped the communication channel between a legal user and the server and recorded the

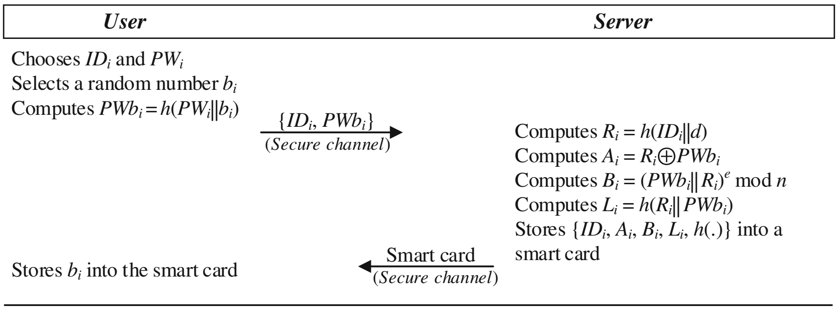


Fig. 2 Registration phase of Giri et al.'s scheme

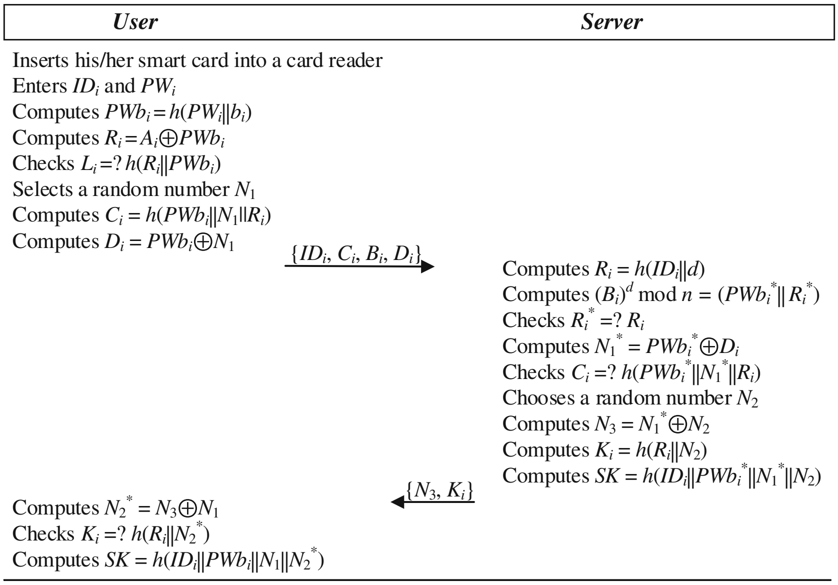


Fig. 3 Login and authentication phases of Giri et al.'s scheme

login request message $\{ID_i, C_i, B_i, D_i\}$. The adversary can login to the server as follows:

Step 1. The adversary sends the eavesdropped login request message $\{ID_i, C_i, B_i, D_i\}$ to the server.

Step 2. Upon receiving the message $\{ID_i, C_i, B_i, D_i\}$, the server computes $R_i = h(ID_i \parallel d)$ and $(B_i)^d \bmod n = (PWb_i^* \parallel R_i^*)$ and checks whether R_i^* is equal to R_i or not. Since they are equal, the server computes $N_1^* = PWb_i^* \oplus D_i$ and checks whether $h(PWb_i^* \parallel N_1^* \parallel R_i)$ is equal to the received C_i or not. Since they are equal, the server authenticates the adversary as a legal user and accepts his/her login request.

Therefore, the adversary can impersonate a legal user and login to the server by replaying an old login request message.

Perfect forward secrecy

Perfect forward secrecy is an important security requirement for security protocols. Perfect forward secrecy ensures that even if an adversary obtains the secret key of one party (e.g., the secret key of the server or the user's password), he/she still cannot compute the previously negotiated session keys [19, 36, 37]. The following demonstrates that Giri et al.'s scheme [13] does not provide perfect forward secrecy.

Suppose an adversary has eavesdropped and recorded the previously transmitted messages $\{ID_i, C_i, B_i, D_i\}$ and $\{K_i, N_3\}$. If the adversary somehow obtains the secret key of the server (d), he/she can compute the previously negotiated session keys as follows:

Step 1. The adversary decrypts B_i with the obtained secret key d as $(B_i)^d \bmod n = (PWb_i^* \parallel R_i^*)$ and computes $N_1^* = PWb_i^* \oplus D_i$ and $N_3 = N_1^* \oplus N_2$.

Step 2. The adversary computes the session key SK as $SK = h(ID_i \parallel PWb_i^* \parallel N_1^* \parallel N_2)$.

Therefore, since in Giri et al.'s scheme [13] disclosure of the server's secret key leads to compromising old session keys, we can conclude that Giri et al.'s scheme does not provide perfect forward secrecy.

Review of Amin and Biswas's scheme

In this section, we briefly review Amin and Biswas's improved authentication and key agreement scheme [12]. Amin and Biswas's scheme [12] includes six phases, i.e., initialization phase, registration phase, login phase, authentication and session key agreement phase, password change phase, and identity change phase. Since the password and identity change phases of Amin and Biswas's scheme is not relevant to our analysis and also the initialization phase of Amin and Biswas's scheme is the same as that of Giri et al.'s scheme (please refer to "Initialization phase"), we only review the following phases of Amin and Biswas's scheme. The notations used in Amin and Biswas's scheme are listed in Table 1.

Registration phase

In this phase, as shown in Fig. 4, a new user can register with the server and obtain a personalized smart card as follows:

Step 1. The user chooses his/her identity ID_i and password PW_i and selects a random number b_i. Then,

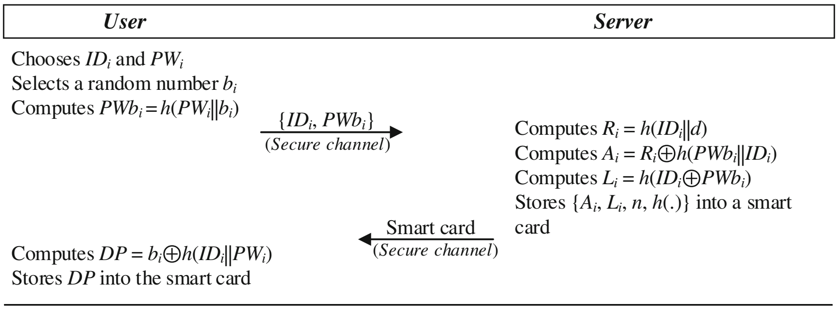


Fig. 4 Registration phase of Amin and Biswas's scheme

the user computes $PWb_i = h(PW_i \parallel b_i)$ and sends a registration request message $\{ID_i, PWb_i\}$ to the server through a secure channel.

Step 2. Upon receiving the registration request message $\{ID_i, PWb_i\}$, the server computes $R_i = h(ID_i \parallel d)$, $A_i = R_i \oplus h(PWb_i \parallel ID_i)$, and $L_i = h(ID_i \oplus PWb_i)$. Then, the server stores $\{A_i, L_i, n, h(\cdot)\}$ into a smart card, and sends the smart card to the user through the secure channel.

Step 3. When the user receives the smart card, he/she computes $DP = b_i \oplus h(ID_i \parallel PW_i)$ and stores DP in the memory of the smart card.

Login phase

When a user wants to login to the server, he/she inserts his/her smart card into the card reader and enters his/her identity ID_i and password PW_i. Then, the smart card computes $b_i = DP \oplus h(ID_i \parallel PW_i)$ and $PWb_i = h(PW_i \parallel b_i)$ and checks whether $h(ID_i \oplus PWb_i)$ is equal to the stored L_i or not. If they are not equal, the smart card terminates the process. Otherwise, the smart card selects a random number N_1 and computes $R_i = A_i \oplus h(PWb_i \parallel ID_i)$, $C_i = h(PWb_i \parallel N_1 \parallel R_i)$, $D_i = h(ID_i \parallel PWb_i) \oplus N_1$, and $B_i = (ID_i \parallel PWb_i \parallel N_1)^e \bmod n$. At last, the smart card sends a message $\{C_i, B_i, D_i\}$ to the server through a public channel.

Authentication and session key agreement phase

In this phase, as shown in Fig. 5, the user and the server check the authenticity of each other and negotiate a session key as follows:

Step 1. Upon receiving the message $\{C_i, B_i, D_i\}$, the server decrypts B_i as $(B_i)^d \bmod n = (ID_i \parallel PWb_i \parallel N_1)$, computes $N_1 = h(ID_i \parallel PWb_i) \oplus D_i$, and checks whether the decrypted N_1 is equal to the computed N_1 or not. If they are not equal, the server terminates the session; otherwise, it computes $R_i = h(ID_i \parallel d)$ and checks whether $h(PWb_i \parallel N_1 \parallel R_i)$ is equal to the received C_i or not. If they are not equal, the server terminates the session. Otherwise, the server authenticates the user, accepts his/her login request, selects a random number N_2, and computes $N_3 = N_1 \oplus N_2$ and $K_i = h(R_i \parallel N_2)$. At last, the server sends a message $\{N_3, K_i\}$ to the user through the public channel.

Step 2. Upon receiving the message $\{N_3, K_i\}$, the user computes $N_2 = N_3 \oplus N_1$ and checks whether $h(R_i \parallel N_2)$ is equal to the received K_i or not. If they are not equal, the user terminates the session. Otherwise, the user authenticates the server and computes the session key SK as $SK = h(ID_i \parallel PWb_i \parallel N_1 \parallel N_2)$. Furthermore, the user computes $SKV = h(SK \parallel ID_i)$ and sends a message $\{SKV\}$ to the server for verification of the session key.

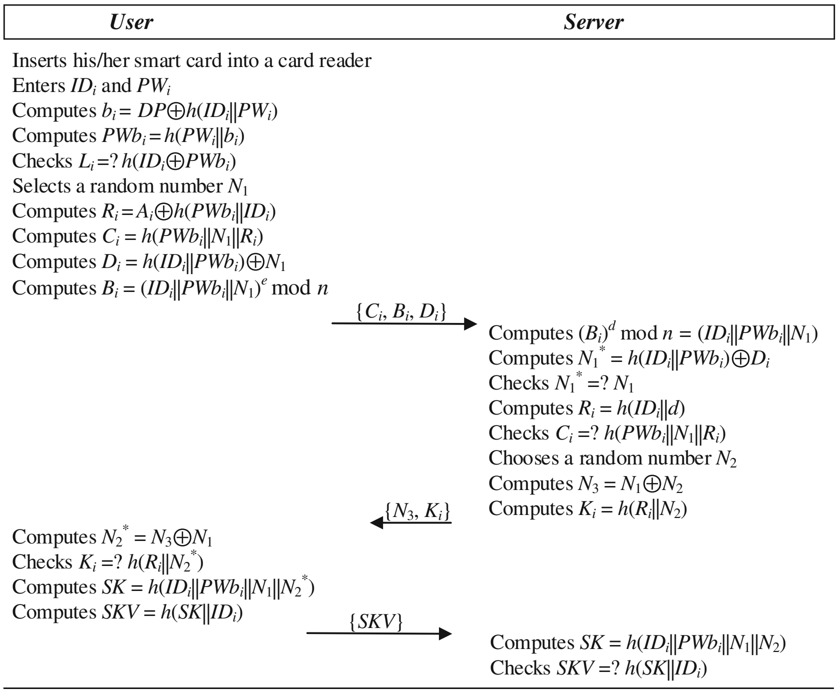
Step 3. After receiving the message $\{SKV\}$, the server computes the session key $SK = h(ID_i \parallel PWb_i \parallel N_1 \parallel N_2)$ and checks whether $h(SK \parallel ID_i)$ is equal to the received SKV or not. If they are equal, the server uses the session key SK for securing the communication between itself and the user.

Weaknesses of Amin and Biswas's scheme

Amin and Biswas [12] claimed that their scheme could withstand various security attacks. However, this section demonstrates that their scheme is vulnerable to off-line password guessing attacks and replay attacks and also does not provide perfect forward secrecy. The details are as follows.

Off-line password guessing attacks

Amin and Biswas [12] claimed that even if an adversary can retrieve $\{A_i, L_i, DP, n, h(\cdot)\}$ from a user's smart card, he/she still cannot guess the user's password, because he/she does not know the secret key of the server (d). However, this section demonstrates that if an adversary steals or finds



Inserts his/her smart card into a card reader
Enters ID_i and PW_i
Computes $b_i = DP \oplus h(ID_i \| PW_i)$
Computes $PWb_i = h(PW_i \| b_i)$
Checks $L_i =? h(ID_i \oplus PWb_i)$
Selects a random number N_1
Computes $R_i = A_i \oplus h(PWb_i \| ID_i)$
Computes $C_i = h(PWb_i \| N_1 \| R_i)$
Computes $D_i = h(ID_i \| PWb_i) \oplus N_1$
Computes $B_i = (ID_i \| PWb_i \| N_1)^e \bmod n$

$\{C_i, B_i, D_i\}$

Computes $(B_i)^d \bmod n = (ID_i \| PWb_i \| N_1)$
Computes $N_1^* = h(ID_i \| PWb_i) \oplus D_i$
Checks $N_1^* =? N_1$
Computes $R_i = h(ID_i \| d)$
Checks $C_i =? h(PWb_i \| N_1 \| R_i)$
Chooses a random number N_2
Computes $N_3 = N_1 \oplus N_2$
Computes $K_i = h(R_i \| N_2)$

$\{N_3, K_i\}$

Computes $N_2^* = N_3 \oplus N_1$
Checks $K_i =? h(R_i \| N_2^*)$
Computes $SK = h(ID_i \| PWb_i \| N_1 \| N_2^*)$
Computes $SKV = h(SK \| ID_i)$

$\{SKV\}$

Computes $SK = h(ID_i \| PWb_i \| N_1 \| N_2)$
Checks $SKV =? h(SK \| ID_i)$

Fig. 5 Login and authentication phases of Amin and Biswas's scheme

a user's smart card, he/she can guess the user's password as follows:

Step 1. The adversary retrieves $\{A_i, L_i, DP, n, h(\cdot)\}$ from the memory of the smart card by using the methods proposed in [38, 39].

Step 2. The adversary selects a pair (ID_i^*, PW_i^*) from two separate dictionaries D_{ID} and D_{PW}. Then, the adversary computes $b_i^* = h(ID_i^* \parallel PW_i^*) \oplus DP$, $PWb_i^* = h(PW_i^* \parallel b_i^*)$, and $L_i^* = h(ID_i^* \oplus PWb_i^*)$ and checks whether the computed L_i^* is equal to the retrieved L_i or not. If they are equal, it implies that the adversary has selected the right pair (ID_i^*, PW_i^*); otherwise, the adversary repeats this step until he/she succeeds.

The off-line password guessing attack is feasible because due to the low entropy nature of the user's identity and password, the adversary can enumerate all the pairs (ID_i^*, PW_i^*) in the Cartesian product $D_{ID} \times D_{PW}$ within polynomial time [40–45].

Replay attacks

Suppose an adversary has eavesdropped the communication channel between a legal user and the server and recorded a previous login request message $\{C_i, B_i, D_i\}$. The adversary can login to the server by sending the eavesdropped login request message $\{C_i, B_i, D_i\}$ to the server. When the server receives the message $\{C_i, B_i, D_i\}$, it decrypts B_i as $(B_i)^d \bmod n = (ID_i \parallel PWb_i \parallel N_1)$, computes $N_1 = h(ID_i \parallel PWb_i) \oplus D_i$, and checks whether the decrypted N_1 is equal to the computed N_1 or not. Since they are equal, the server computes $R_i = h(ID_i \parallel d)$ and checks whether $h(PWb_i \parallel N_1 \parallel R_i)$ is equal to the received C_i or not. Since they are equal, the server authenticates the adversary as a legal user and accepts his/her login request. Furthermore, the server selects a random number N_2, computes $N_3 = N_1 \oplus N_2$ and $K_i = h(R_i \parallel N_2)$, and sends a message $\{N_3, K_i\}$ to the user (adversary). Although the adversary cannot compute the session key SK, he/she is successful as long as the server accepts the login request. Hence, since the server authenticated the adversary as the legal user and accepted his/her login request, the adversary ignores the received message $\{N_3, K_i\}$.

Therefore, since an adversary can impersonate a legal user and login to the server by replaying an old login request message, we can conclude that Amin and Biswas's scheme [12] is vulnerable to replay attacks.

Perfect forward secrecy

As mentioned before, the perfect forward secrecy is an important security requirement for authentication and key agreement protocols. This section demonstrates that similar to Giri et al.'s scheme [13], Amin and Biswas's scheme [12] also does not provide perfect forward secrecy.

Suppose an adversary has eavesdropped and recorded the previously transmitted messages $\{C_i, B_i, D_i\}$ and $\{K_i, N_3\}$. If the adversary somehow obtains the secret key of the server (d), he/she can compute the previously established session keys as follows:

Step 1. The adversary decrypts B_i with the obtained secret key d as $(B_i)^d \bmod n = (ID_i \parallel PWb_i \parallel N_1)$ and computes $N_1 = h(ID_i \parallel PWb_i) \oplus D_i$ and $N_2 = N_3 \oplus N_1$.

Step 2. The adversary computes the session key SK as $SK = h(ID_i \parallel PWb_i \parallel N_1 \parallel N_2)$.

Therefore, since divulgence of the server's secret key compromises the secrecy of the old session keys, it can be claimed that Amin and Biswas's scheme [12] does not provide perfect forward secrecy.

The proposed scheme

In order to overcome the security weaknesses of Giri et al.'s scheme [13] and Amin and Biswas's scheme [12], a secure and efficient authentication and key agreement scheme for TMISs is proposed in this section. The proposed scheme consists of four phases: initialization phase, registration phase, login and authentication phase, and password change phase. The notations used in the proposed scheme are listed in Table 2 and the phases are illustrated in the following subsections.

Table 2 Notations used in the proposed scheme

Symbol	*Description*
E	An elliptic curve
P	The base point of the elliptic curve
xP	The point multiplication defined as $xP = \underbrace{P+P+\ldots+P}_{x \text{ times}}$
PW_i	The user's password
ID_i	The user's identity
CID_i	The user's dynamic identity
s	The server's secret key
$\parallel$	The concatenation operation
$\oplus$	The exclusive-or (XOR) operation
SK	The shared session key between the user and the server
T_1, T_2	Two timestamps
ΔT	The maximum transmission delay
$E_k(\cdot)/D_k(\cdot)$	The symmetric encryption/decryption with the key k
$h(\cdot)$	A secure one-way hash function

Initialization phase

In this phase, the server chooses an elliptic curve E [50] and selects a point P with the large order n over the elliptic curve as the base point. Then, the server selects a random number $s \in_R Z_p^*$ as its secret key and a secure one-way hash function $h(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^l$, where l is the length of the output. Finally, the server publishes $\{E, n, P, h(\cdot)\}$ and keeps s secretly.

Registration phase

As shown in Fig. 6, the user registration process is as follows:

Step 1. The user chooses his/her identity ID_i and password PW_i, selects a random number b_i, and computes $PWb_i = h(PW_i \parallel b_i)$. At last, the user sends a registration request message $\{ID_i, PWb_i\}$ to the server through a secure channel.

Step 2. Upon receiving the message $\{ID_i, PWb_i\}$, the server checks whether ID_i exists in its database or not. If it exists, the server asks the user to choose another identity. Otherwise, the server chooses a random number r, computes $R_i = h(ID_i \parallel s)$, $A_i = R_i \oplus h(ID_i \parallel PWb_i)$, and $CID_i = E_s(ID_i \parallel r)$, stores ID_i in its database and $\{A_i, CID_i, E, P, n, h(\cdot)\}$ into a smart card, and sends the smart card to the user through the secure channel.

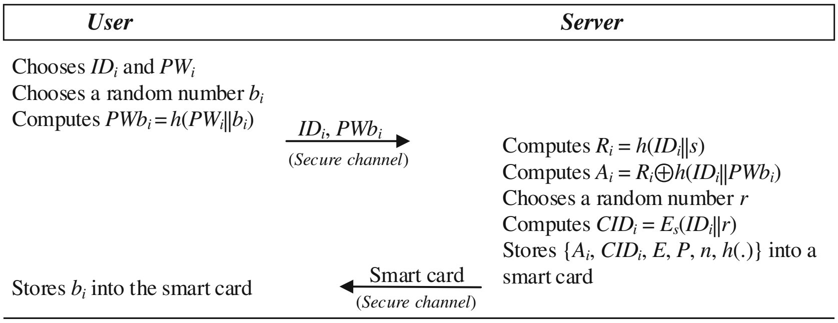
Step 3. When the user receives the smart card, he/she stores the random number b_i in the memory of the smart card.

Login and authentication phase

In this phase, as shown in Fig. 7, the user and the server authenticate each other and negotiate a session key as follows:

Step 1. The user inserts his/her smart card into the card reader and enters his/her identity ID_i and password PW_i. Then, the smart card selects a random number $k_1 \in_R Z_p^*$ and computes $K_1 = k_1 P$, $R_i = A_i \oplus h(ID_i \parallel h(PW_i \parallel b_i))$, and $V_1 = h(ID_i \parallel K_1 \parallel R_i \parallel T_1)$, where T_1 is the current timestamp. At last, the smart card sends a login request message $\{CID_i, K_1, V_1, T_1\}$ to the server through a public channel.

Step 2. Upon receiving the message $\{CID_i, K_1, V_1, T_1\}$, the server checks the freshness of the timestamp T_1 by checking the condition $T_2 - T_1 \ ? \le \Delta T$, where T_2 is the time when the server receives the login request message $\{CID_i, K_1, V_1, T_1\}$ and ΔT denotes the maximum transmission delay. If it is not fresh, the server ignores the received login request message. Otherwise, the server computes $D_s(CID_i) = (ID_i \parallel r)$ and checks whether the received V_1 is equal to $h(ID_i \parallel K_1 \parallel$



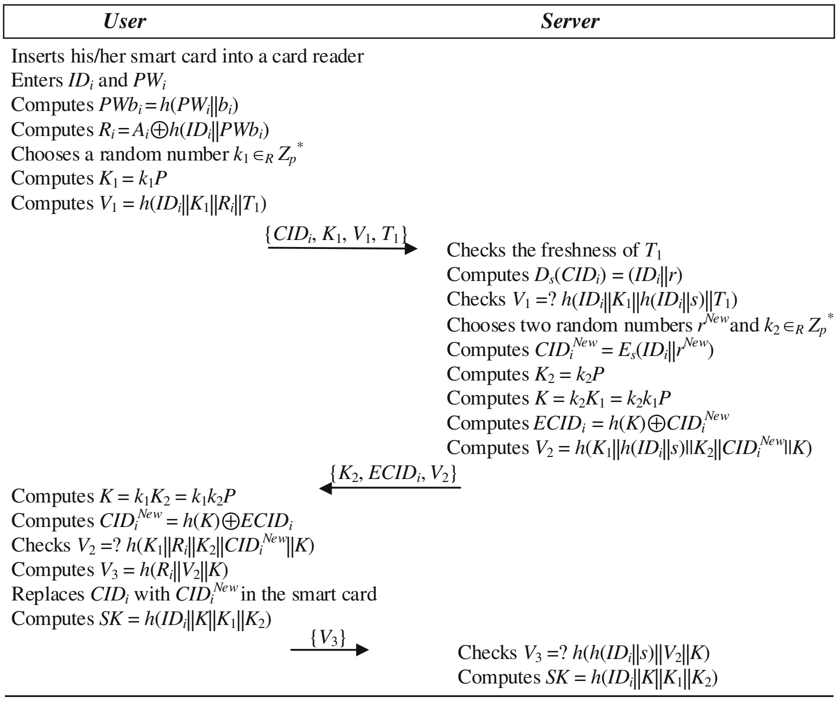
Chooses ID_i and PW_i
Chooses a random number b_i
Computes $PWb_i = h(PW_i \| b_i)$
ID_i, PWb_i (*Secure channel*)
Computes $R_i = h(ID_i \| s)$
Computes $A_i = R_i \oplus h(ID_i \| PWb_i)$
Chooses a random number r
Computes $CID_i = E_s(ID_i \| r)$
Stores $\{A_i, CID_i, E, P, n, h(.)\}$ into a smart card
Smart card (*Secure channel*)
Stores b_i into the smart card

Fig. 6 Registration phase of the proposed scheme

$h(ID_i \parallel s) \parallel T_1)$ or not. If they are not equal, the server terminates the session. Otherwise, the server chooses two random numbers r^{New} and $k_2 \in_R Z_p^*$, computes $CID_i^{New} = E_s(ID_i \parallel r^{New})$, $K_2 = k_2 P$, $K = k_2 K_1$, $ECID_i = h(K) \oplus CID_i^{New}$, and $V_2 = h(K_1 \parallel h(ID_i \parallel s) \parallel K_2 \parallel CID_i^{New} \parallel K)$, and sends a challenge message $\{K_2, ECID_i, V_2\}$ to the user through the public channel. It should be noted that the server does not send the value of CID_i^{New} in plaintext through the public channel. Therefore, an adversary cannot establish a link between the exchanged messages over the public channel and the user (smart card) who sent/received them. In fact, the server sends the new dynamic identity of the user (CID_i^{New}) in a protected manner as $ECID_i = h(K) \oplus CID_i^{New}$ in order to withstand off-line password guessing attacks as discussed in "Password guessing attacks".

Step 3. When the user receives the message $\{K_2, ECID_i, V_2\}$, he/she computes $K = k_1 K_2$ and $CID_i^{New} = h(K) \oplus ECID_i$ and checks whether $h(K_1 \parallel R_i \parallel K_2 \parallel CID_i^{New} \parallel K)$ is equal to the received V_2 or not. If they are not equal, the user terminates the session. Otherwise, the user authenticates the server, computes $V_3 = h(R_i \parallel V_2 \parallel K)$, replaces CID_i with CID_i^{New} in the smart card, and sends a response message $\{V_3\}$ to the server through the public channel. Furthermore, the user computes the session key SK as $SK = h(ID_i \parallel K \parallel K_1 \parallel K_2)$.

Step 4. After receiving the message $\{V_3\}$, the server checks whether $h(h(ID_i \parallel s) \parallel V_2 \parallel K)$ is equal to the received V_3 or not. If they are not equal, the server terminates the session; otherwise, the server authenticates the user and computes the session key SK as $SK = h(ID_i \parallel K \parallel K_1 \parallel K_2)$.



Inserts his/her smart card into a card reader
Enters ID_i and PW_i
Computes $PWb_i = h(PW_i \| b_i)$
Computes $R_i = A_i \oplus h(ID_i \| PWb_i)$
Chooses a random number $k_1 \in_R Z_p^*$
Computes $K_1 = k_1 P$
Computes $V_1 = h(ID_i \| K_1 \| R_i \| T_1)$
$\{CID_i, K_1, V_1, T_1\}$
Checks the freshness of T_1
Computes $D_s(CID_i) = (ID_i \| r)$
Checks $V_1 =? h(ID_i \| K_1 \| h(ID_i \| s) \| T_1)$
Chooses two random numbers r^{New} and $k_2 \in_R Z_p^*$
Computes $CID_i^{New} = E_s(ID_i \| r^{New})$
Computes $K_2 = k_2 P$
Computes $K = k_2 K_1 = k_2 k_1 P$
Computes $ECID_i = h(K) \oplus CID_i^{New}$
Computes $V_2 = h(K_1 \| h(ID_i \| s) \| K_2 \| CID_i^{New} \| K)$
$\{K_2, ECID_i, V_2\}$
Computes $K = k_1 K_2 = k_1 k_2 P$
Computes $CID_i^{New} = h(K) \oplus ECID_i$
Checks $V_2 =? h(K_1 \| R_i \| K_2 \| CID_i^{New} \| K)$
Computes $V_3 = h(R_i \| V_2 \| K)$
Replaces CID_i with CID_i^{New} in the smart card
Computes $SK = h(ID_i \| K \| K_1 \| K_2)$
$\{V_3\}$
Checks $V_3 =? h(h(ID_i \| s) \| V_2 \| K)$
Computes $SK = h(ID_i \| K \| K_1 \| K_2)$

Fig. 7 Login and authentication phase of the proposed scheme

Password change phase

When a user wants to change his/her password, he/she inserts his/her smart card into the card reader and keys in his/her identity ID_i, his/her current password PW_i, and a new password PW_i^{New}. Then, the smart card and the server perform the following steps.

Step 1. This step is the same as Step 1 in "Login and authentication phase".

Step 2. This step is the same as Step 2 in "Login and authentication phase".

Step 3. After receiving the message $\{K_2, ECID_i, V_2\}$, the smart card computes $K = k_1K_2$ and $CID_i^{New} = h(K) \oplus ECID_i$ and checks whether $h(K_1 \parallel R_i \parallel K_2 \parallel CID_i^{New} \parallel K)$ is equal to the received V_2 or not. If they are not equal, the smart card stops the process. Otherwise, the smart card computes A_i^{New} as $A_i^{New} = A_i \oplus h(ID_i \parallel h(PW_i \parallel b_i)) \oplus h(ID_i \parallel h(PW_i^{New} \parallel b_i)) = R_i \oplus h(ID_i \parallel h(PW_i \parallel b_i)) \oplus h(ID_i \parallel h(PW_i \parallel b_i)) \oplus h(ID_i \parallel h(PW_i^{New} \parallel b_i)) = R_i \oplus h(ID_i \parallel h(PW_i^{New} \parallel b_i))$ and replaces CID_i and A_i with CID_i^{New} and A_i^{New}, respectively.

Security analysis

In this section, the security of the proposed scheme is analyzed. In the following, first the correctness of the proposed scheme is proved and then resistance of the proposed scheme against various attacks is examined.

Authentication proof based on GNY logic

In this section, GNY (Gong-Needham-Yalahom) logic [46] is employed to prove the correctness of the proposed scheme. In order to analyze the proposed scheme, the following rules of GNY logic [46] are used, where the index numbers are based on [46]. Table 3 summarizes the notations employed in this section.

- $T1: \dfrac{A \triangleleft *X}{A \triangleleft X}$
- $T3: \dfrac{A \triangleleft \{X\}_K,\ A \ni K}{A \triangleleft X}$
- $R1: \dfrac{A \mid\equiv \phi(X)}{A \mid\equiv \phi(X, Y),\ A \mid\equiv \phi(F(X))}$,
- $R2: \dfrac{A \mid\equiv \phi(X),\ A \ni K}{A \mid\equiv \phi(\{X\}_K),\ A \mid\equiv \phi(\{X\}_K^{-1})}$
- $R5: \dfrac{A \mid\equiv \phi(X),\ A \ni X}{A \mid\equiv \phi(H(X))}$
- $R6: \dfrac{A \ni H(X)}{A \mid\equiv \phi(X)}$

Table 3 GNY-logic notations

Symbol	*Description*
U_i	A user
S	The server
$*X$	X is not originated here
$A \ni X$	A possesses X
$A \triangleleft X$	A is told X
$A \mid\sim X$	A once conveyed X
$A \mid\equiv X$	A believes X
$A \mid\equiv \#(X)$	A believes that X is fresh
$A \mid\equiv \phi(X)$	A believes that X is recognizable
$A \mid\Rightarrow X$	A has jurisdiction over X
$A \mid\equiv A \overset{K}{\leftrightarrow} B$	A believes that K is a suitable secret for A and B
$\{X\}_K / \{X\}_K^{-1}$	Conventional encryption/decryption of X with key K
$H(X)$	A one-way function of X
(X, Y)	Conjunction of X and Y

- $P1: \dfrac{A \triangleleft X}{A \ni X}$
- $P4: \dfrac{A \ni X}{A \ni H(X)}$
- $P5: \dfrac{A \ni F(X, Y),\ A \ni X}{A \ni Y}$
- $F1: \dfrac{A \mid\equiv \#(X)}{A \mid\equiv \#(X, Y),\ A \mid\equiv \#F(X)}$
- $I1: \dfrac{A \triangleleft *\{X\}_K,\ A \ni K,\ A \mid\equiv A \overset{K}{\leftrightarrow} B,\ A \mid\equiv \phi(X),\ A \mid\equiv \#(X, K)}{A \mid\equiv B \mid\sim X,\ A \mid\equiv B \mid\sim \{X\}_K,\ A \mid\equiv B \ni K}$
- $I3: \dfrac{A \triangleleft *H(X, < S >),\ A \ni (X, S),\ A \mid\equiv A \overset{S}{\leftrightarrow} B,\ A \mid\equiv \#(X, S)}{A \mid\equiv B \mid\sim (X, < S >),\ A \mid\equiv B \mid\sim H(X, < S >)}$
- $I6: \dfrac{A \mid\equiv B \mid\sim X,\ A \mid\equiv \#(X)}{A \mid\equiv B \ni X}$
- $J1: \dfrac{A \mid\equiv B \mid\Rightarrow C,\ A \mid\equiv B \mid\equiv C}{A \mid\equiv C}$
- $J2: \dfrac{A \mid\equiv B \mid\Rightarrow B \mid\equiv *,\ A \mid\equiv B \mid\sim (X \leadsto C),\ A \mid\equiv \#(X)}{A \mid\equiv B \mid\equiv C}$

According to GNY logic, the proposed scheme must satisfy the following goals, which are categorized into three aspects:

- *Message content authentication:*
    - **Goal 1**: $S \mid\equiv \phi(\{ID_i, r\}_s, k_1P, H(ID_i, k_1P, R_i, T_1), T_1)$
    - **Goal 2**: $U_i \mid\equiv \phi(k_2P, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), F(H(k_1k_2P), \{ID_i, r^{New}\}_s))$
    - **Goal 3**: $S \mid\equiv \phi(H(R_i, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), k_1k_2P))$
- *Message origin authentication:*
    - **Goal 4**: $U_i \mid\equiv S \mid\sim (k_2P, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), F(H(k_1k_2P), \{ID_i, r^{New}\}_s))$

  - **Goal 5**: $S \mid\equiv U_i \mid\sim H(R_i, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), k_1k_2P)$

- *Session key establishment:*
  - **Goal 6**: $U_i \mid\equiv S \mid\equiv (U_i \xleftrightarrow{K} S)$
  - **Goal 7**: $U_i \mid\equiv (U_i \xleftrightarrow{K} S)$
  - **Goal 8**: $S \mid\equiv U_i \ni K$
  - **Goal 9**: $S \mid\equiv U_i \mid\equiv (U_i \xleftrightarrow{K} S)$

In order to analyze the proposed scheme using GNY logic, the proposed scheme is specified as follows:

Message 1: $U_i \rightarrow S : (\{ID_i, r\}_s, k_1P, H(ID_i, k_1P, R_i, T_1), T_1)$

Message 2: $S \rightarrow U_i : (k_2P, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), F(H(k_1k_2P), \{ID_i, r^{New}\}_s))$

Message 3: $U_i \rightarrow S : H(R_i, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), k_1k_2P)$

In addition, the following assumptions are made to analyze the proposed scheme:

- $A_1 : S \ni s$
- $A_2 : S \mid\equiv \phi(ID_i)$
- $A_3 : S \ni R_i$
- $A_4 : U_i \ni k_1$
- $A_5 : U_i \ni R_i$
- $A_6 : S \ni k_2$
- $A_7 : U_i \mid\equiv (U_iU_i \xleftrightarrow{R_i} S)$
- $A_8 : U_i \mid\equiv \#(k_1)$
- $A_9 : S \mid\equiv (U_i \xleftrightarrow{K} S)$
- $A_{10} : S \mid\equiv \#(k_2)$
- $A_{11} : U_i \mid\equiv S \mid\Rightarrow (U_i \xleftrightarrow{K} S)$

According to the rules of GNY logic, the proposed scheme is analyzed as follows:

According to *Message* 1, the following is obtained:

O_1: $S \lhd \ast(\ast\{ID_i, r\}_s, \ast k_1P, \ast H(ID_i, k_1P, R_i, T_1), \ast T_1)$

By applying the rule $T1$ to O_1, the following is obtained:

O_2: $S \lhd (\{ID_i, r\}_s, k_1P, H(ID_i, k_1P, R_i, T_1), T_1)$

Based on O_2 ($S \lhd \{ID_i, r\}_s$) and A_1, the rule $T3$ is applied to obtain:

O_3: $S \lhd (ID_i, r)$

According to O_2, O_3, and the rule $P1$, the following is obtained:

O_4: $S \ni ID_i, r, k_1P, T_1$

According to A_2 and the rule $R1$, the following are obtained:

O_5: $S \mid\equiv \phi(ID_i, r)$
O_6: $S \mid\equiv \phi(ID_i, k_1P, R_i, T_1)$

Based on O_5 and A_1, the rule $R2$ is applied to obtain:

O_7: $S \mid\equiv \phi(\{ID_i, r\}_s)$

According to O_6, O_4, and A_3, the rule $R5$ is applied to deduce:

O_8: $S \mid\equiv \phi(H(ID_i, k_1P, R_i, T_1))$

According to O_7, O_8, and the rule $R1$, the following is obtained:

O_9: $S \mid\equiv \phi(\{ID_i, r\}_s, k_1P, H(ID_i, k_1P, R_i, T_1), T_1)$ (**Goal 1**)

According to *Message* 2, the following is obtained:

O_{10}: $U_i \lhd \ast(\ast k_2P, \ast H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), \ast \mathrm{F}(H(k_1k_2P), \{ID_i, r^{New}\}_s))$

By applying the rule $T1$ to O_{10}, the following is obtained:

O_{11}: $U_i \lhd (k_2P, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), F(H(k_1k_2P), \{ID_i, r^{New}\}_s))$

By applying the rule $P1$ to O_{11}, the following is obtained:

O_{12}: $U_i \ni k_2P, F(H(k_1k_2P), \{ID_i, r^{New}\}_s)$

Based on O_{12} ($U_i \ni k_2P$) and A_4, the following is deduced:

O_{13}: $U_i \ni k_1k_2P$

By applying the rule $P4$ to O_{13}, the following is obtained:

O_{14}: $U_i \ni H(k_1k_2P)$

According to O_{12} ($U_i \ni F(H(k_1k_2P), \{ID_i, r^{New}\}_s)$) and O_{14}, the rule $P5$ is applied to obtain:

O_{15}: $U_i \ni \{ID_i, r^{New}\}_s$

Since U_i posses k_1 (according to A_4), U_i can compute k_1P and thus the following can be deduced:

O_{16}: $U_i \ni k_1P$

By applying the rule $P4$ to O_{16}, the following is obtained:

O_{17}: $U_i \ni H(k_1P)$

Based on O_{17} and the rule $R6$, the following is obtained:

O_{18}: $U_i \mid\equiv \phi(k_1P)$

According to O_{18} and the rule $R1$, the following is obtained:

O_{19}: $U_i \mid\equiv \phi(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P)$

Based on O_{19}, O_{16}, O_{15}, O_{13}, O_{12} ($U_i \ni k_2P$), A_5, and the rule $R5$, the following is obtained:

O_{20}: $U_i \mid\equiv \phi H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P)$

According to O_{20} and the rule $R1$, the following is obtained:

O_{21}: $U_i \mid\equiv \phi(k_2P, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), F(\mathrm{H}(k_1k_2P), \{ID_i, r^{New}\}_s))$ **(Goal 2)**

According to *Message* 3, the following is obtained:

O_{22}: $S \triangleleft *H(R_i, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), k_1k_2P)$

By applying the rule $T1$ to O_{22}, the following is obtained:

O_{23}: $S \triangleleft H(R_i, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), k_1k_2P)$

Based on O_{23} and the rule $P1$, the following is obtained:

O_{24}: $S \ni H(R_i, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), k_1k_2P)$

Based on O_{24} and the rule $R6$, the following is obtained:

O_{25}: $S \mid\equiv \phi(R_i, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), k_1k_2P)$

Since, according to *Message* 2, *S* sends $H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P)$ to U_i, the following can be deduced:

O_{26}: $S \ni H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P)$

Based on O_4 ($S \ni k_1P$) and A_6, the following can be deduced:

O_{27}: $S \ni k_1k_2P$

Based on O_{25}, O_{26}, O_{27}, A_3, and the rule $R5$, the following is obtained:

O_{28}: $S \mid\equiv \phi H(R_i, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), k_1k_2P)$ **(Goal 3)**

According to O_{10}, O_{12}, O_{13}, A_5, A_7, and A_8, rules $F1$ and $I3$ are applied to obtain:

O_{29}: $U_i \mid\equiv S \mid\sim (k_2P, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), F(H(k_1k_2P), \{ID_i, r^{New}\}_s))$ **(Goal 4)**

Based on O_{22}, A_3, O_{26}, O_{27}, A_9, and A_{10}, rules $F1$ and $I3$ are applied to obtain:

O_{30}: $S \mid\equiv U_i \mid\sim H(R_i, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), k_1k_2P)$ **(Goal 5)**

O_{31}: $S \mid\equiv U_i \mid\sim (R_i, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), k_1k_2P)$

Based on O_{31}, A_{10}, $K = k_1k_2P = k_2k_1P$, and the rules $F1$ and $I6$, the following is obtained:

O_{32}: $S \mid\equiv U_i \ni K$ **(Goal 8)**

According to GNY logic, it is assumed that U_i believes that *S* is honest and competent, $U_i \mid\equiv S \mid\Rightarrow S \mid\equiv *$. Hence, based on $U_i \mid\equiv S \mid\sim (k_2P, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), F(H(k_1k_2P), \{ID_i, r^{New}\}_s)) \rightsquigarrow S \mid\equiv U_i \overset{k_1k_2P}{\longleftrightarrow} S$ (O_{29}), A_8, and $K = k_1k_2P = k_2k_1P$, rules $F1$ and $J2$ are applied to obtain:

O_{33}: $U_i \mid\equiv S \mid\equiv (U_i \overset{K}{\longleftrightarrow} S)$ **(Goal 6)**

According to O_{33} and A_{11}, the rule $J1$ is applied to obtain:

O_{34}: $U_i \mid\equiv (U_i \overset{K}{\longleftrightarrow} S)$ **(Goal 7)**

According to GNY logic, it is assumed that *S* believes that U_i is honest and competent, $S \mid\equiv U_i \mid\Rightarrow U_i \mid\equiv *$. Hence, based on $S \mid\equiv U_i \mid\sim H(R_i, H(k_1P, R_i, k_2P, \{ID_i, r^{New}\}_s, k_1k_2P), k_1k_2P) \rightsquigarrow U_i \mid\equiv U_i \overset{k_1k_2P}{\longleftrightarrow} S$ (O_{30}), A_{10}, and $K = k_1k_2P = k_2k_1P$, rules $F1$ and $J2$ are applied to obtain:

O_{35}: $S \mid\equiv U_i \mid\equiv (U_i \overset{K}{\longleftrightarrow} S)$ **(Goal 9)**

Formal security verification using AVISPA tool

In this subsection, the widely accepted and used AVISPA tool [47] is used to prove the security of the proposed scheme. AVISPA is a push-button tool for automated validation of security protocols that integrates four different back-ends, which employ various automatic analysis methods. In order to analyze a protocol using the AVISPA, the protocol and its intended security properties should be described and specified by the High Level Protocol Specification Language (HLPSL) [48], which is a role-oriented language. The AVISPA translates the HLPSL specification of the protocol into the Intermediate Format (IF) using the hlpsl2if translator. Then, the intended security properties of the protocol can be formally validated by analyzing the IF codes using each of the four back-ends of the AVISPA.

In order to formally validate the proposed scheme using the AVISPA, the registration and the login and authentication phases of the proposed scheme are specified in HLPSL. The HLPSL specifications of the user and server roles in the proposed scheme are shown in Figs. 8 and 9, respectively.

In addition to the user and server roles, two other roles, namely the session role and the environment role should be specified in HLPSL. As shown in Fig. 10, the session role describes a session of the protocol by describing the interactions between the user and the server. The environment role describes a composition of one or more sessions and contains the intruder knowledge and the global constants. Figure 11 shows that in the environment role, the intruder, which is denoted by i, can play the role of the user and the server.

```
role user(Ui,S:agent,
      H, F: hash_func,
      P: text,
      SND, RCV: channel (dy))

played_by Ui
def=
  local State:nat,
     IDi,PWi,PWBi,Bi,CIDi,ECIDi,CIDiNew,Ri:text,
     K1,K2,K1p,K2p,K,Ai,V1,V2,V3,T1:text
     SK, Kuis: symmetric_key

const ui_s_k1,s_ui_k2,g0,g1,g2,g3,g4:protocol_id
init  State := 0
transition

%Registration phase

1. State = 0  /\ RCV(start) =|>
      State':= 1
      /\ Bi' := new()
      /\ PWBi' := H(PWi.Bi)
      /\ SND({IDi.PWBi'}_Kuis)
      /\ secret({PWi, Bi}, g0, Ui)

2. State = 1 /\ RCV({Ai.CIDi}_Kuis) =|>
%Login and authentication phase
      State':= 2
      /\ Ri' := xor(Ai, H(IDi.H(PWi.Bi)))
      /\ K1' := new()
      /\ K1p' := F(K1'.P)
      /\ T1' := new()
      /\ V1' := H(IDi.K1p'.Ri'.T1')
      /\ SND(CIDi.K1p'.V1'.T1')
      /\ witness(Ui,S,ui_s_k1,K1')
      /\ secret(K1',g1,Ui)
      /\ secret(IDi,g2,{Ui,S})

2. State = 2 /\ RCV(K2p.ECIDi.V2) =|>
      State':= 3
      /\ K' := F(K1.K2p)
      /\ CIDiNew' := xor(ECIDi,H(K'))
      /\ V3' := H(Ri.V2.K')
      /\ SK' := H(IDi.K'.K1p.K2p)
      /\ SND(V3')
      /\ secret(K',g3,{Ui,S})
      /\ secret(SK',g4,{Ui,S})
      /\ request(S,Ui,s_ui_k2,K2)
end role
```

Fig. 8 The HLPSL specification of the user

```
role server(Ui,S:agent,
       H, F: hash_func,
       P: text,
       SND, RCV: channel (dy))
played_by S
def=
local State :nat,
      IDi,PWBi,CIDi,ECIDi,CIDiNew,SS,T1,Ai:text,
      K1,K2,K1p,K2p,K,V1,V2,V3,R,Ri,RNew:text
      SK, Kuis: symmetric_key

const ui_s_k1,s_ui_k2,g5,g6: protocol_id

init  State := 0
transition

%Registration phase

1. State = 0 /\ RCV({IDi.PWBi}_Kuis) =|>
      State':= 1
      /\ Ri' := H(IDi.SS)
      /\ Ai' := xor(Ri', H(IDi.PWBi))
      /\ R' := new()
      /\ CIDi' := {IDi.R'}_SS
      /\ SND({Ai'.CIDi'}_Kuis)
      /\ secret({R', SS}, g5, S)
%Login and authentication phase

1. State = 1 /\ RCV(CIDi.K1p.V1.T1) =|>
      State':= 2
      /\ RNew' := new()
      /\ K2' := new()
      /\ CIDiNew' := {IDi.RNew'}_SS
      /\ K2p' := F(K2.P)
      /\ K' := F(K2.K1p)
      /\ ECIDi' := xor(CIDiNew', H(K'))
      /\ V2' := H(K1p.H(IDi.SS).K2p'.CIDiNew'.K')
      /\ SND(K2p'.ECIDi'.V2')
      /\ witness(S,Ui,s_ui_k2,K2)
      /\ secret(K2',g6,S)

2. State = 2 /\ RCV(V3) =|>
      State':= 3
      /\ SK' := H(IDi.K.K1p.K2p)
      /\ request(Ui,S,ui_s_k1,K1)
end role
```

Fig. 9 The HLPSL specification of the server

After describing the user, the server, the session, and the environment roles, the intended security properties and goals of the proposed scheme are specified as shown in Fig. 12. In the goal section, `secrecy_of g0`, where `g0` is a protocol id for the statement `secret({PWi, Bi}, g0, Ui)`, means that the user's password PW_i and the random number b_i are kept secret to the user. The goal `secrecy_of g1`, where `g1` is a protocol id for the statement `secret(K1',g1,Ui)`, means that the random number k_1 is kept secret to the user. The goal `secrecy_of g2`, where `g2` is a reference to the statement `secret (IDi, g2, {Ui, S})`, indicates that the real identity of the user (ID_i) is kept secret to the user and the server. The goal `secrecy_of g3`, where `g3` is a reference to the statement `secret (K', g3, {Ui, S})`, means that the key $K = k_1k_2P$ is kept secret to the user and the server. The goal `secrecy_of g4`, where `g4` refers to the statement `secret(SK',g4,{Ui,S})`, indicates that the session key SK is kept secret to the user and the server. The goal `secrecy_of g5`, where `g5` refers to the statement `secret({R', SS}, g5, S)`, means that the secret key of the server (s) and the random number r are kept secret to the server (SS and R denote the server's secret key (s) and the random number r, respectively). The goal `secrecy_of g6`, where `g6` is a reference to the statement `secret(K2',g6,S)`, indicates that the random number k_2 is kept secret to the server. The goal `authentication_on ui_s_k1` means that the

```
role session(Ui, S : agent,
             H, F   : hash_func,
             P      : text)
def=
  local
    SND1,RCV1,SND2,RCV2: channel(dy)

  composition
    user(Ui,S,H,F,P,SND1,RCV1)
    /\ server(Ui,S,H,F,P,SND2,RCV2)

end role
```

Fig. 10 The HLPSL specification of the session role

user selects a random number k_1 and the server authenticates the user after receiving k_1 from the messages from the user. The goal `authentication_on s_ui_k2` indicates that the server selects a random number k_2 and the user authenticates the server after receiving k_2 from the messages from the server.

The results of analyzing the proposed scheme using the AVISPA with the widely-accepted OFMC (On-the fly Model-Checker) back-end [49] are shown in Fig. 13. The results confirm that the stated security goals were satisfied for a bounded number of sessions as specified in the environment role. Therefore, the proposed scheme is safe and can withstand passive and active attacks.

Discussion on the possible attacks

This section demonstrates that the proposed scheme withstands insider attacks, replay attacks, password guessing attacks, and impersonation attacks and provides perfect forward secrecy, user anonymity, and known-key security.

User anonymity

```
role environment()
def=
 const ui, s, i : agent,
       p        : text,
       h,f      : hash_func,
      ui_s_k1,s_ui_k2,g0,g1,g2,g3,g4,g5,g6:protocol_id
intruder_knowledge = {ui, s, h, f, p}
composition

        session(ui, s, h, f, p)
        /\ session(i, ui, h, f, p)
        /\ session(ui, i, h, f, p)
end role
```

Fig. 11 The HLPSL specification of the environment role

```
goal

secrecy_of g0
secrecy_of g1
secrecy_of g2
secrecy_of g3
secrecy_of g4
secrecy_of g5
secrecy_of g6
authentication_on ui_s_k1
authentication_on s_ui_k2

end goal
```

Fig. 12 The HLPSL specification of the security goals

Generally, user anonymity includes two aspects, i.e., the protection of the user's real identity and the untraceability of the user. In the proposed scheme, the user's real identity ID_i is never transmitted over the public channel. If the adversary gets the user's login request message $\{CID_i, K_1, V_1, T_1\}$, he/she cannot reveal the user's real identity ID_i, because it is encrypted with the server's secret key s as $CID_i = E_s(ID_i \parallel r)$ and the adversary does not know the server's secret key s. Therefore, it is impossible for the adversary to reveal the user's real identity ID_i from the login and authentication messages.

Besides, in each new session, the new random numbers k_1 and k_2 and timestamp T_1 are used to generate the communication messages, and the smart card information CID_i is updated as $CID_i{}^{New} = E_s(ID_i \parallel r^{New})$ after each

```
% OFMC
% Version of 2006/02/13
SUMMARY
    SAFE
DETAILS
    BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
   /cdrom/avispa-1.1/testsuite/results/Medical.if
GOAL
   as_specified
BACKEND
   OFMC
COMMENTS
STATISTICS
   prseTime: 0.00s
   searchTime: 0.04s
   visitedNodes: 8 nodes
   depth: 3 plies
```

Fig. 13 The output of the OFMC back-end

Table 4 Notations used in the performance analysis

Symbol	*Description*
T_E	Time for performing an exponentiation operation
T_{PM}	Time for performing an elliptic curve point multiplication operation
T_{SED}	Time for performing a symmetric encryption/decryption operation
T_H	Time for performing a hash function operation
T_X	Time for performing an exclusive-or operation

successful login. Therefore, since all values of the communication messages $\{CID_i, K_1, V_1, T_1\}$, $\{K_2, ECID_i, V_2\}$, and $\{V_3\}$ in one session are different from those of any other sessions, an adversary cannot relate the session with a specific user and the proposed scheme can ensure untraceability of the user.

Therefore, it can be said that the proposed scheme can provide the property of user anonymity.

Password guessing attacks

There are two kinds of password guessing attacks, i.e., online password guessing attack and off-line password guessing attack, where in the last one the adversary tries to verify the correctness of the guessed password by using the previously transmitted messages or (and) the stolen smart card information. We first discuss the off-line password guessing attack.

Suppose an adversary steals a smart card of a user and retrieves $\{A_i, CID_i, E, P, n, b_i, h(\cdot)\}$ from the memory of the smart card, where $A_i = h(ID_i \parallel s) \oplus h(ID_i \parallel h(PW_i \parallel b_i))$ and $CID_i = E_s(ID_i \parallel r)$. The adversary cannot derive the user's identity ID_i from CID_i, because he/she does not know the server's secret key s, with the same reason, he/she cannot guess the right ID_i and PW_i from A_i. Therefore, the adversary cannot guess the password from the information on the stolen smart card.

The adversary may use of the previously transmitted messages $\{CID_i, K_1, V_1, T_1\}$, $\{K_2, ECID_i, V_2\}$, and $\{V_3\}$ to guess the password. However, since CID_i changes after each successful login, and the random numbers k_1 and k_2 and timestamp T_1 are fresh in each session, all values in the login and authentication messages of a user are different in each session (see "User anonymity"). Hence, the adversary cannot link the eavesdropped login and authentication messages to the corresponding user (or smart card), i.e., the adversary cannot distinguish which messages belong to the stolen smart card. Therefore, the adversary has no way to verify the correctness of the guessed password PW_i by using the previously transmitted login and authentication messages. It should be noted that the dynamic identity CID_i that is stored on the smart card, has never been transmitted over the public channel previously and the server submitted it in a protected manner as $ECID_i = {CID_i}^{New} \oplus h(k_1 k_2 P)$ to the user in the previous session. In fact, the dynamic identity CID_i that is stored on the smart card is not included in any previously transmitted messages.

From the above analysis, it can be said that the proposed scheme could withstand off-line password guessing attacks. Besides, for the online password guessing attack, it is well known that it can be defeated by limiting the number of continuous failed login requests [4, 7, 8, 19].

Therefore, the proposed scheme could withstand password guessing attacks.

Insider attacks

During the registration phase of the proposed scheme, each user sends his/her masked password $PWb_i = h(PW_i \parallel b_i)$ to the server. Hence, since the hash function is one-way and the random number b_i is unknown to anyone except the user, a privileged user of the server has no chance to obtain or guess the user's password PW_i. Therefore, the proposed scheme could withstand insider attacks.

Replay attacks

An adversary may replay a previous login request message $\{CID_i, K_1, V_1, T_1\}$ to the server. However, the server could detect a replay attack by checking the freshness of the timestamp T_1 as $T_2 - T_1 \ ?\le \Delta T$, where T_2 is the time when the server receives the message $\{CID_i, K_1, V_1, T_1\}$ and ΔT is the maximum transmission delay. The adversary may also replay a previous challenge message $\{K_2, ECID_i, V_2\}$ to the user. However, since the smart card has generated a new random number k_1 in this session, the user could detect a replay attack by checking $h(k_1 P \parallel R_i \parallel K_2 \parallel CID_i^{New} \parallel k_1 K_2) =? V_2$. Therefore, the proposed scheme could withstand replay attacks.

Impersonation attacks

In the proposed scheme, an adversary cannot produce a valid login request message $\{CID_i, K_1, V_1, T_1\}$, where $CID_i = E_s(ID_i \parallel r)$ and $V_1 = h(ID_i \parallel K_1 \parallel h(ID_i \parallel s) \parallel T_1)$, because he/she does not know the server's secret key s and the user's identity ID_i. The adversary may steal a smart card and retrieve $\{A_i, CID_i, b_i\}$ from the memory of the smart card, where $A_i = h(ID_i \parallel s) \oplus h(ID_i \parallel h(PW_i \parallel b_i))$ and $CID_i = E_s(ID_i \parallel r)$. However, since the adversary does not know the user's password PW_i, he/she cannot obtain $h(ID_i \parallel s)$ and thus he/she cannot produce a valid login request message $\{CID_i, K_1, V_1, T_1\}$. Therefore, no one can impersonate a legal user. Moreover, the adversary cannot produce a valid challenge message $\{K_2, ECID_i, V_2\}$, where

Table 5 Comparison of the proposed scheme with the related schemes

Comparison criteria			Scheme			
			Bin Muhaya[11]	Giri et al. [13]	Amin and Biswas [12]	The proposed
Computational cost	Registration phase	Cost	$3T_H + 3T_X$	$1T_E + 3T_H + 1T_X$	$5T_H + 3T_X$	$1T_{SED} + 3T_H + 1T_X$
		Time	$1.5ms$	$523.5ms$	$2.5ms$	$10.2ms$
	Login and authentication phase	Cost	$2T_E + 12T_H + 3T_X$	$1T_E + 9T_H + 5T_X$	$2T_E + 15T_H + 7T_X$	$4T_{PM} + 2T_{SED} + 14T_H + 3T_X$
		Time	$1050ms$	$526.5ms$	$1051.5ms$	$276.7ms$
Security properties	Resist password guessing attacks		No	No	No	Yes
	Resist replay attacks		Yes	No	No	Yes
	Resist impersonation attacks		Yes	Yes	Yes	Yes
	Resist privileged insider attacks		Yes	No	Yes	Yes
	Provide perfect forward secrecy		No	No	No	Yes
	Provide mutual authentication		Yes	Yes	Yes	Yes
	Provide known-key security		Yes	Yes	Yes	Yes
	Provide key agreement		Yes	Yes	Yes	Yes
	Preserve user privacy		Yes	No	Yes	Yes

$V_2 = h(K_1 \parallel h(ID_i \parallel s) \parallel K_2 \parallel h(k_2K_1) \oplus ECID_i \parallel k_2K_1)$, because he/she does not know the server's secret key s. Therefore, no one can impersonate a legal server.

Perfect forward secrecy

In the proposed scheme, the user and the server compute the session key SK as $SK = h(ID_i \parallel k_1k_2P \parallel k_1P \parallel k_2P)$, where k_1 and k_2 are random numbers chosen by the user and the server, respectively. Knowing the server's secret key s or the user's password PW_i does not help an adversary to compute previously established session keys, because the secret values s and PW_i are not utilized to compute session keys. If an adversary wants to obtain an old session key, he/she has to compute k_1k_2P. However, since the adversary does not know k_1 or k_2 and cannot derive them from k_1P and k_2P (due to the hardness of ECDLP [50]), he/she cannot compute k_1k_2P. Therefore, the proposed scheme provides perfect forward secrecy.

Know-key security

In the proposed scheme, if an adversary somehow obtains a session key $SK = h(ID_i \parallel K \parallel K_1 \parallel K_2)$, he/she still cannot compute other session keys due to the randomness of $K(= k_1k_2P)$, $K_1(= k_1P)$, and $K_2(= k_2P)$. Therefore, the proposed scheme provides know-key security.

Performance analysis

In this section, the performance and security of the proposed scheme are compared with those of Amin and Biswas's scheme [12], Giri et al.'s scheme [13], and Bin Muhaya's scheme [11].

For convenience to evaluate the computational cost, some notations are defined in Table 4. According to [51, 52], the computation time of an exponentiation operation, an elliptic curve point multiplication operation, a hash function operation, and a symmetric encryption/decryption operation is 0.522 s, 0.063075 s, 0.0005 s, and 0.0087 s, respectively. Moreover, it is assumed that the time for executing an exclusive-or (XOR) operation is negligible.

In the proposed scheme, one symmetric encryption operation, one exclusive-or operation, and three hash function operations are required for the registration process. Hence, the computational cost of the registration phase of the proposed scheme is $1T_{SED} + 3T_H + 1T_X$, which is equivalent to 10.2 ms. Besides, four elliptic curve point multiplication operations, one symmetric encryption operation, fourteen hash function operations, one symmetric decryption operation, and three exclusive-or operations are required for the login and authentication processes. Hence, the computational cost of the login and authentication phase of the proposed scheme is $4T_{PM} + 2T_{SED} + 14T_H + 3T_X$, which is equivalent to 276.7 ms.

Table 5 demonstrates the comparisons among the proposed scheme, Amin and Biswas's scheme [12], Giri et al.'s scheme [13], and Bin Muhaya's scheme [11] in terms of the computational costs and security properties. Moreover, Fig. 14 shows the running times of the proposed scheme, Amin and Biswas's scheme [12], Giri et al.'s scheme [13], and Bin Muhaya's scheme [11].

From Table 5, it is clear that the proposed scheme is more efficient than Amin and Biswas's scheme [12], Giri et al.'s scheme [13], and Bin Muhaya's scheme [11]. In the login and authentication phase, the proposed scheme is about 3.79, 1.9, and 3.8 times faster than the schemes of Bin Muhaya [11], Giri et al. [13], and Amin and Biswas [12], respectively. Moreover, the schemes proposed by Amin and Biswas [12], Giri et al. [13], and Bin Muhaya's scheme [11] are vulnerable to password guessing attacks, whereas the proposed scheme is secure against password guessing attacks. Amin and Biswas's scheme [12] and Giri et al.'s scheme [13] both are vulnerable to replay attacks, whereas the proposed scheme resists replay attacks. Amin and Biswas's scheme [12], Giri et al.'s scheme [13], and Bin Muhaya's scheme [11] do not provide perfect forward

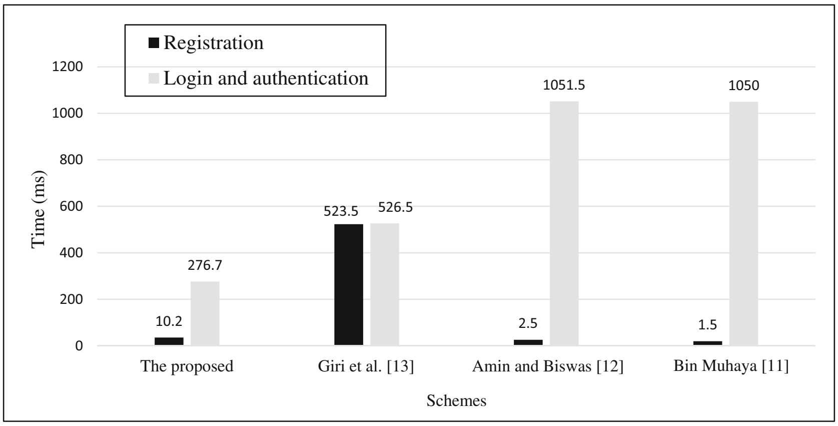


Fig. 14 Running times of different schemes

secrecy, whereas the proposed scheme provides perfect forward secrecy. Giri et al.'s scheme [13] is susceptible to privileged insider attacks and does not preserve user privacy, whereas the proposed scheme resists privileged insider attacks and preserves the privacy of the user. It is worth to mention that in comparison with the other ECC-base authentication schemes existing in the literature, the proposed scheme needs fewer scalar multiplication operations. Since the scalar multiplication operation (the elliptic curve point multiplication) is the main (time-consuming) operation in elliptic curve cryptosystems, the performance of the proposed scheme is much better than the other ECC-base authentication schemes. Therefore, the proposed is more suitable for practical applications.

Conclusion

In this paper, we have demonstrated some possible attacks on the authentication schemes proposed by Giri et al. and Amin and Biswas. We also have shown that these two schemes do not provide perfect forward secrecy. Then, in order to improve the security and efficiency, we have proposed a novel authentication and key agreement scheme for TMISs. We have employed the GNY logic to show the correctness of the proposed scheme. We also have simulated the proposed scheme for the formal verification using the well-known AVISPA tool. Security analysis demonstrates that the proposed scheme not only could withstand various attacks, but also could provide perfect forward secrecy, user anonymity, and know-key security. According to the performance analysis, the proposed scheme has a better performance than the previous schemes. Therefore, the proposed scheme is more suitable for TMISs.

References

1. Zhang, K., Yang, K., Liang, X., Su, Z., Shen, X., and Luo, H. H., Security and privacy for mobile healthcare networks: from a quality of protection perspective. *IEEE Wireless Communications* 22(4):104–112, 2015.
2. He, D., Kumar, N., Chen, J., Lee, C. C., Chilamkurti, N., and Yeo, S. S., Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. *Multimedia Systems* 21(1):49–60, 2015.
3. He, D., and Zeadally, S., Certificateless public auditing scheme for cloud-assisted wireless body area networks. *IEEE Systems Journal*, 2015. doi:10.1109/JSYST.2015.2428620.
4. Mir, O., Munilla, J., and Kumari, S. *Efficient anonymous authentication with key agreement protocol for wireless medical sensor networks*, 2015. doi:10.1007/s12083-015-0408-1.
5. Mir, O., and Nikooghadam, M., A secure biometrics based authentication with key agreement scheme in telemedicine networks for e-health services. *Wirel Pers Commun* 83(4):2439–2461, 2015.
6. He, D., Zeadally, S., Kumar, N., and Lee, J. H. *Anonymous authentication for wireless body area networks with provable security*, 2016. doi:10.1109/JSYST.2016.2544805.
7. Nikooghadam, M., Jahantigh, R., and Arshad, H., A lightweight authentication and key agreement protocol preserving user anonymity. *Multimedia Tools and Applications*, 2016. doi:10.1007/s11042-016-3704-8.
8. Arshad, H., and Nikooghadam, M., Security analysis and improvement of two authentication and key agreement schemes for session initiation protocol. *J Supercomput* 71(8):3163–3180, 2015.
9. Zhang, L., Zhu, S., and Tang, S., Privacy protection for telecare medicine information systems using a chaotic map-based three-factor authenticated key agreement scheme. *IEEE Journal of Biomedical and Health Informatics*, 2016. doi:10.1109/JBHI.2016.2517146.
10. Liu, W., Xie, Q., Wang, S., and Hu, B., An improved authenticated key agreement protocol for telecare medicine information system. *SpringerPlus* 5(1):1–16, 2016.
11. Bin Muhaya, F. T., Cryptanalysis and security enhancement of Zhu's authentication scheme for Telecare medicine information system. *Security and Communication Networks* 8(2):149–158, 2015.
12. Amin, R., and Biswas, G. P., An improved rsa based user authentication and session key agreement protocol usable in TMIS. *Journal of Medical Systems* 39(8):1–14, 2015.
13. Giri, D., Maitra, T., Amin, R., and Srivastava, P. D., An Efficient and Robust RSA-Based Remote User Authentication for Telecare Medical Information Systems. *Journal of medical systems* 39(1):1–9, 2015.
14. Lamport, L., Password authentication with insecure communication. *Commun ACM* 24(11):770–772, 1981.
15. Lennon, R., Matyas, S., and Mayer, C., Cryptographic authentication of time-invariant quantities. *IEEE Trans Commun* 6:773–777, 1981.
16. Yen, S., and Liao, K., Shared authentication token secure against replay and weak key attack. *Inf Process Lett*, 78–80, 1997.
17. He, D., Wang, H., Wang L, Shen, J., and Yang, X., Efficient certificateless anonymous multi-receiver encryption scheme for mobile devices. *Soft Computing*, 2016. doi:10.1007/s00500-016-2231-x.
18. He, D., Zhang, M., and Xu, B., Insecurity of an Efficient Identity-Based Proxy Signature in the Standard Model. *The Computer Journal* 58(10):2507–2508, 2015.
19. Arshad, H., and Nikooghadam, M., An efficient and secure authentication and key agreement scheme for session protocol using ECC. *Multimed Tools Appl* 75(1):181–197, 2016.
20. He, D., Kumar, N., and Lee, J. H., Privacy-preserving data aggregation scheme against internal attackers in smart grids. *Wireless Networks* 22(2):491–502, 2016.
21. Ramaki, A. A., Amini, M., and Atani, R. E., RTECA: Real time episode correlation algorithm for multi-step attack scenarios detection. *Computers & Security* 49:206–219, 2015.
22. He, D., Kumar, N., Shen, H., and Lee, J. H., One-to-many authentication for access control in mobile pay-TV systems. *Science China-Information Sciences* 59(5):1–14, 2016. doi:10.1007/s11432-015-5469-5.
23. Hwang, M., and Li, L., A new remote user authentication scheme using smart cards. *IEEE Trans Consum Electron* 46(1):28–30, 2000.
24. Sharif, A., Mollaeefar, M., and Nazari, M., A novel method for digital image steganography based on a new three-dimensional chaotic map. *Multimedia Tools and Applications*, 2016. doi:10.1007/s11042-016-3398-y.
25. Mollaeefar, M., Sharif, A., and Nazari, M., A novel encryption scheme for colored image based on high level chaotic maps. *Multimedia Tools and Applications*, 2015. doi:10.1007/s11042-015-3064-9.

26. Chan, C.-K., and Cheng, L.-M., Cryptanalysis of a remote user authentication scheme using smart cards. *IEEE Trans Consum Electron* 46(4):992–993, 2000.
27. Sun, H.-M., An efficient remote use authentication scheme using smart cards. *IEEE Trans Consum Electron* 46(4):958–961, 2000.
28. Chien, H.-Y., Jan, J.-K., and Tseng, Y.-M., An efficient and practical solution to remote authentication: smart card. *Comput Secur* 21(4):372–375, 2002.
29. Ku, W., Chen, C., and Lee, H., Cryptanalysis of a variant of peyravian-zunic's password authentication scheme. *IEICE Trans Commun* E86-B(5):1682–1684, 2003.
30. Yoon, E., Ryu, E., and Yoo, K., Further improvement of an efficient password based remote user authentication scheme using smart cards. *IEEE Trans Consum Electron* 50(2):612–614, 2004.
31. Wang, X., Zhang, W., Zhang, J., and Khan, M., Cryptanalysis and improvement on two efficient remote user authentication scheme using cards. *Comput Stand Interfaces* 29(5):507–512, 2007.
32. Hsieh, W., and Leu, J., Exploiting hash functions to intensify the remote user authentication scheme. *Comput Secur* 31(6):791–798, 2012.
33. Wang, D., Ma, C., Wang, P., and Chen, Z., Robust smart card based password authentication scheme against smart card security breach. IACR Cryptology ePrint Archive. Retrieved from eprint.iacr.org/2012/439.eps, 2012.
34. Chang, Y., Tai, W., and Chang H, Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update. *Int J Commun Syst*, 2013. doi:10.1002/dac.2552.
35. Kumari, S., Gupta, M. K., Khan, M. K., and Li, X., An improved timestamp-based password authentication scheme: comments, cryptanalysis, and improvement. *Secur Commun Netw* 7(11):1921–1932, 2014.
36. He, D., Zeadally, S., Xu, B., and Huang, X., An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Transactions on Information Forensics and Security* 10(12):2681–2691, 2015.
37. He, D., Kumar, N., and Lee, J. H., Secure pseudonym-based near field communication protocol for the consumer internet of things. *IEEE Transactions on Consumer Electronics* 61(1):56–62, 2015.
38. Kocher, P., Jaffe, J., and Jun, B., Differential power analysis. In: Proceedings of Advances in Cryptology, Santa Barbara, CA, USA, 1666, pp. 788–797, 1999.
39. Messerges, T. S., Dabbish, E. A., and Sloan, R. H., Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers* 51(5):541–552, 2002.
40. Wang, D., and Wang, P., Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks. *Ad Hoc Networks* 20:1–15, 2014.
41. Ma, C. G., Wang, D., and Zhao, S. D., Security flaws in two improved remote user authentication schemes using smart cards. *International Journal of Communication Systems* 27(10):2215–2227, 2014.
42. Klein, D. V., Foiling the cracker: a survey of, and improvements to, password security. In: Proceedings of the 2^{nd} USENIX Security Workshop, Anaheim, CA, USA, August, pp. 5–14, 1990.
43. Kumari, S., Li, X., Wu, F., Das, A. K., Arshad, H., and Khan, M. K., A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. *Future Generation Computer Systems* 63:56–75, 2016.
44. Bonneau, J., The science of guessing: analyzing an anonymized corpus of 70 million passwords. In: 33^{th} IEEE Symposium on Security and Privacy (S&P 2012), IEEE Computer Society, San Francisco, CA, USA, May, pp. 538–552, 2012.
45. Islam, S. H., Design and analysis of an improved smartcard-based remote user password authentication scheme. *International Journal of Communication Systems*, 2014. doi:10.1002/dac.2793.
46. Gong, L., Needham, R., and Yahalom, R., Reasoning about belief in cryptographic protocols. In: Proc 1990 IEEE Computer Society Symp. Research in Security and Privacy, pp. 234–246, 1990.
47. Armando, A., Basin, D., Cuellar, J., Rusinowitch, M., and Vigan, L., AVISPA: Automated Validation of Internet Security Protocols and Applications. *ERCIM News*, 64, 2006.
48. Chevalier, Y., Compagna, L., Cuellar, J., Hankes, D. P., Mantovani, J., Modersheim, S., and Vigneron, L., A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols. In: Proc. SAPS'04. Austrian Computer Society, 2004.
49. Basin, D., Modersheim, S., and Vigano, L., OFMC: A symbolic model checker for security protocols. *International Journal of Information Security* 4(3):181–208, 2005.
50. Hankerson, D., Menezes, A., and Vanstone, S. *Guide to elliptic curve cryptography*. New York: Springer, 2004.
51. He, D., Kumar, N., Khan, M. K., and Lee, J. H., Anonymous Two-factor Authentication for Consumer Roaming Service in Global Mobility Networks. *IEEE Transactions on Consumer Electronics* 59(4):811–817, 2013.
52. Jiang, Q., Ma, J., Li, G., and Yang, L., An Efficient Ticket Based Authentication Protocol with Unlinkability for Wireless Access Networks. *Wireless Personal Communications* 77(2):1489–1506, 2014.