Hindawi
Security and Communication Networks
Volume 2019, Article ID 4656281, 13 pages
https://doi.org/10.1155/2019/4656281

Research Article

A Secure and Efficient ECC-Based Anonymous Authentication Protocol

Feifei Wang, Guoai Xu, and Lize Gu

Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Guoai Xu; xga@bupt.edu.cn

Received 16 August 2018; Accepted 24 June 2019; Published 20 August 2019

Academic Editor: Dimitrios Geneiatakis

Nowadays, remote user authentication protocol plays a great role in ensuring the security of data transmission and protecting the privacy of users for various network services. In this study, we discover two recently introduced anonymous authentication schemes are not as secure as they claimed, by demonstrating they suffer from offline password guessing attack, desynchronization attack, session key disclosure attack, failure to achieve user anonymity, or forward secrecy. Besides, we reveal two environment-specific authentication schemes have weaknesses like impersonation attack. To eliminate the security vulnerabilities of existing schemes, we propose an improved authentication scheme based on elliptic curve cryptosystem. We use BAN logic and heuristic analysis to prove our scheme provides perfect security attributes and is resistant to known attacks. In addition, the security and performance comparison show that our scheme is superior with better security and low computation and communication cost.

1. Introduction

With the dramatic increase of network attacks and privacy leakage, it is extremely important for various network services to identify the authenticity of communicating party and protect the privacy of users in insecure environment. As a basic defense strategy for numerous network services, the authentication protocol is aimed at solving these security issues. It has been used in various areas, such as e-banking, e-health, wireless sensor networks, and internet of things [1–3]. Authentication protocol generally provides three useful functionalities, that is, mutual authentication, user anonymity, and session key agreement.

Recently, there have been a great number of authentication schemes introduced and some evaluation metrics developed [4–8]. In 2012, Wang et al. [9] presented a robust authentication protocol that is resistant to known attacks, but the protocol has low efficiency, as it requires a number of modular exponentiation operations. Besides, they presented a comprehensive evaluation criteria for smart card based password authentication protocols. Madhusudhan and Mittal [10] defined the security requirements and desirable properties an authentication protocol should fulfil and provide. Kim–Kim [11] introduced an efficient dynamic identity authentication scheme, in which a synchronization mechanism is adopted to achieve user anonymity. In 2014, Islam et al. [12] introduced an anonymous authentication scheme based on elliptic curve cryptosystem (ECC). Huang et al. [13] proposed an anonymous authentication protocol based on RSA cryptosystem. In 2015, Wang et al. [14] demonstrated Kim–Kim's scheme and its similar schemes that employ the same synchronization mechanism cannot withstand desynchronization attack and introduced an ElGamal cryptosystem based scheme to overcome this threat. In 2016, Nikooghadam et al. [15] presented an efficient authentication protocol using symmetric key cryptosystem and claimed the protocol is resistant to various known attacks. Jung et al. [16] proposed a symmetric key cryptosystem based authentication and key agreement protocol for wireless sensor networks. In 2017, Luo et al. [17] proved that Islam et al.'s scheme is susceptible to insider attack and offline password guessing attack and introduced a new ECC-based scheme for improvement. But their scheme suffers from session key disclosure attack and offline guessing password attack. Xiong et al. [18] proved Jung et al.'s protocol fails to achieve forward secrecy and suffers from smart card loss attack and introduced an improved scheme based on hash chain technique. Xie et al. [19] proposed a provably secure authentication protocol using

ECC. Unfortunately, the protocol is inefficient in detection of wrong identity and password. Amin et al. [20] demonstrated that Huang et al.'s scheme cannot resist offline password guessing attack and forgery attack and presented an anonymous RSA cryptosystem based scheme for improvement.

Although a great number of research works have been done on authentication protocols [21–35], new authentication schemes still have various security weaknesses [22–29, 32–34]. Offline guessing password attack and forward secrecy attack are two of the most common security weaknesses.

One prominent issue of authentication protocol is security against offline guessing password attack. In authentication scheme, the verification value is essential to check the validity of inputted password and implement local password updates. But the introduced verification value probably leads to offline password guessing attack, even server impersonation attack, user impersonation attack, and man-in-the-middle attack. To solve this problem, Wang et al. [35] introduced an effective solution by integrating "fuzzy-verifier" with "honeywords".

On the other hand, forward secrecy is a matter of concern to authentication protocol. Numerous authentication schemes proposed recently achieve many desirable features but fail to preserve forward secrecy. Forward secrecy attack is a security vulnerability that damages the whole server system. Halevi and Krawczyk [36] demonstrated that, for any key exchange scheme, perfect forward secrecy can be achieved through using Diffie-Hellman key exchange.

1.1. Our Contributions. We cryptanalyze several representative schemes in the paper. Firstly, we point out Amin et al.'s scheme [20] suffers from offline guessing attack, user impersonation attack and fails to provide forward secrecy. Next, we reveal Nikooghadam et al.'s protocol [15] suffers from offline guessing attack, man-in-the-middle attack, session key disclosure attack, server impersonation attack, desynchronization attack, user impersonation attack, and fails to preserve forward secrecy and user anonymity. Then we point out Mishra et al.'s session initiation protocol [33] fails to provide forward secrecy. In addition, we demonstrate Hsieh et al.'s authentication scheme for wireless communication [34] suffers from offline password guessing attack and impersonation attack.

To eliminate these security vulnerabilities, we propose an improved authentication protocol based on elliptic curve cryptosystem. We use BAN logic and informal analysis to prove the completeness and security of our scheme. Furthermore, we give the performance and security comparison of related schemes. The results show that our scheme is more practical.

1.2. Structure of the Paper. Section 2 gives the cryptanalysis of Amin et al.'s scheme. Section 3 is the cryptanalysis of Nikooghadam et al.'s scheme. Section 4 is the cryptanalysis of two environment-specific authentication schemes. Section 5 gives the proposed scheme. Section 6 is the security analysis of our scheme. And Section 7 gives the security and performance comparison of related schemes. Section 8 is a conclusion.

Table 1: Notations.

Symbol	Description
U_i	i^{th} user
S	Remote server
$\mathscr{A}$	Malicious adversary
ID_i	Identity of user U_i
PW_i	Password of user U_i
x	Master key of S
P	A generator P of elliptic curve group E_p
T_i, T_S	The current timestamp of U_i, S
T_1, T_2, T_3	The current timestamp
$E_{Key}()/D_{Key}()$	Symmetric encryption/ decryption algorithm with key Key
SK	Session key between U_i and S
$h(\cdot)$	Hash function
$\|$	The string concatenation operation
$\oplus$	The bitwise XOR operation
$\longrightarrow$	A public communication channel
$\Longrightarrow$	A secure communication channel
ID_{HA}	Identity of home agent
ID_{FA}	Identity of the foreign agent

We elaborate the notations of this paper in Table 1.

2. Cryptanalysis of Amin Et Al.'s Scheme

In this section, we describe Amin et al.'s scheme and reveal its security vulnerabilities.

2.1. Review of Amin Et Al.'s Scheme. Amin et al.'s scheme consists of the following three phases (see Figure 1).

2.1.1. Initialization Phase. S chooses two big prime numbers u, v and calculates $m = u \times v$. Next, S selects another prime number e, where $1 < e < (u-1)(v-1)$. And d is calculated according to $ed \equiv 1 \bmod (u-1)(v-1)$. S publishes public parameters $\{m, e\}$ and keeps $\{d, u, v\}$ as secret.

2.1.2. Registration Phase. In this phase, U_i submits his identity information to S with the purpose of obtaining the access permission.

Step 1. U_i picks ID_i and PW_i freely. U_i calculates $PWR_i = h(PW_i \| r_i)$, where r_i is a nonce. Afterwards, $\{ID_i, PWR_i\}$ is transmitted to S via a secure channel.

Step 2. Upon receiving $\{ID_i, PWR_i\}$, S calculates $A_i = h(d \| ID_i)$, $B_i = h(A_i \| PWR_i \| ID_i)$, $Y_i = A_i \oplus h(PWR_i \| ID_i)$.

U_i/Smart card	Public channel	Server S
U_i selects ID_i, PW_i, random number r_i Computes $PWR_i = h(PW_i \parallel r_i)$	$\{ID_i, PWR_i\}$ ⟹ smart card ⟸	Computes $A_i = h(d \parallel ID_i)$ $B_i = h(A_i \parallel PWR_i \parallel ID_i)$ $Y_i = A_i \oplus h(PWR_i \parallel ID_i)$
Stores r_i in smart card		Stores $\{B_i, Y_i, m, e\}$ in a smart card
Inputs ID_i^*, PW_i^* Computes $PWR_i^* = h(PW_i^* \parallel r_i)$		Computes $\left(ID_i' \parallel D_i' \parallel N_1'\right) = L_i{}^d \bmod m$
$A_i^* = Y_i \oplus h(PWR_i^* \parallel ID_i^*)$	$\{L_i, Y_i, T_i\}$ ⟶	$A_i' = h(d \parallel ID_i')$
$B_i^* = h(A_i^* \parallel PWR_i^* \parallel ID_i^*)$		$[h(PWR_i^* \parallel ID_i^*)]' = Y_i \oplus A_i'$
Checks $B_i^* \stackrel{?}{=} B_i$		$D_i'' = h(A_i' \parallel [h(PWR_i^* \parallel ID_i^*)]' \parallel T_i \parallel N_1')$
Chooses a random number N_1 Computes $D_i = h(A_i^* \parallel h(PWR_i^* \parallel ID_i^*) \parallel T_i \parallel N_1)$ $L_i = (ID_i^* \parallel D_i \parallel N_1)^e \bmod m$		Checks $D_i'' \stackrel{?}{=} D_i'$ Chooses a random number N_2
	$\{X_i, Z_i, T_s\}$ ⟵	Computes $X_i = h(N_2 \parallel A_i')$ $Z_i = N_2 \oplus N_1'$ $SK = h(N_1' \parallel A_i' \parallel N_2)$
Computes $N_2' = Z_i \oplus N_1$ $X_i' = h(N_2' \parallel A_i^*)$		
Checks $X_i' \stackrel{?}{=} X_i$ Computes $SK = h(N_1 \parallel A_i^* \parallel N_2')$		

Figure 1: Amin et al.'s scheme.

S stores $\{B_i, Y_i, m, e\}$ in a smart card. The smart card is transmitted to U_i via a secure channel.

Step 3. U_i stores r_i in the smart card.

2.1.3. Login and Authentication Phase. In this phase, U_i delivers a login request message to S. S verifies the legitimacy of the message and sends back a response.

Step 1. U_i attaches the smart card to a terminal and inputs ID_i^* and PW_i^*. The smart card calculates $PWR_i^* = h(PW_i^* \parallel r_i)$, $A_i^* = Y_i \oplus h(PWR_i^* \parallel ID_i^*)$, $B_i^* = h(A_i^* \parallel PWR_i^* \parallel ID_i^*)$, and checks whether $B_i^* = B_i$. If the equation holds, perform next step.

Step 2. The smart card computes $D_i = h(A_i^* \parallel h(PWR_i^* \parallel ID_i^*) \parallel T_i \parallel N_1)$, $L_i = (ID_i^* \parallel D_i \parallel N_1)^e \bmod m$, where N_1 is random number. $\{L_i, Y_i, T_i\}$ is transmitted to S.

Step 3. After receiving $\{L_i, Y_i, T_i\}$, S validates if $T_i' - T_i < \Delta T$, where T_i' is the current time receives $\{L_i, Y_i, T_i\}$ at server end, and ΔT is an accredited maximum transport delay. If it holds, it denotes that T_i is fresh. S computes $L_i{}^d \bmod m$ to derive $(ID_i' \parallel D_i' \parallel N_1')$. Then S calculates $A_i' = h(d \parallel ID_i')$, $[h(PWR_i^* \parallel ID_i^*)]' = Y_i \oplus A_i'$.

Step 4. S calculates $D_i'' = h(A_i' \parallel [h(PWR_i^* \parallel ID_i^*)]' \parallel T_i \parallel N_1')$ and checks $D_i'' \stackrel{?}{=} D_i'$. If the equation holds, S regards U_i as a legitimate user. Otherwise, this protocol aborts.

Step 5. S computes $X_i = h(N_2 \parallel A_i')$, $Z_i = N_2 \oplus N_1'$, $SK = h(N_1' \parallel A_i' \parallel N_2)$, where N_2 is a random number. S sends $\{X_i, Z_i, T_s\}$ to U_i.

Step 6. After receiving $\{X_i, Z_i, T_s\}$, the smart card first checks if T_s is fresh. Next, the smart card calculates $N_2' = Z_i \oplus N_1$, $X_i' = h(N_2' \parallel A_i^*)$, and checks if $X_i' \stackrel{?}{=} X_i$. If the equation holds, S is authenticated by U_i. Then, the smart card calculates $SK = h(N_1 \parallel A_i^* \parallel N_2')$ as session key.

2.2. Cryptanalysis of Amin Et Al.'s Scheme. In this part, we elaborate Amin et al.'s scheme is susceptible to several security attacks.

2.2.1. Offline Password Guessing Attack. The adversary $\mathcal{A}$ extracts $\{B_i, Y_i, m, e, r_i\}$ from the smart card. $\mathcal{A}$ performs offline password guessing attack in the following steps.

Step 1. Choose an identity ID_i^* from the identity dictionary space, and a password PW^* from the password dictionary space.

Step 2. Calculate $PWR_i^* = h(PW_i^* \parallel r_i)$, $A_i^* = Y_i \oplus h(PWR_i^* \parallel ID_i^*)$, $B_i^* = h(A_i^* \parallel PWR_i^* \parallel ID_i^*)$. Compare B_i^* with B_i. If they are equal, it shows that ID_i^* is U_i's real identity, and PW^* is U_i's correct password.

Step 3. Repeat Steps 1-2, until $\mathcal{A}$ finds the real ID_i and PW_i.

2.2.2. User Impersonation Attack. Once $\mathcal{A}$ extracts $\{B_i, Y_i, m, e, r_i\}$ from the smart card and gets ID_i, PW_i via "offline password guessing attack", $\mathcal{A}$ performs user impersonation attack in the following steps.

Step 1. Calculate $PWR_i^* = h(PW_i \parallel r_i)$, $A_i^* = Y_i \oplus h(PWR_i^* \parallel ID_i)$, $B_i^* = h(A_i^* \parallel PWR_i^* \parallel ID_i)$. Obviously, B_i^* is equal to B_i.

Step 2. Select a nonce N_{1_a}. Calculate $D_a = h(A_i^* \parallel h(PWR_i^* \parallel ID_i) \parallel T_a \parallel N_{1_a})$, where T_a is the current timestamp. Calculate $L_a = (ID_i \parallel D_a \parallel N_{1_a})^e \bmod m$. Send $\{L_a, Y_i, T_a\}$ to S.

U_i/Smart card	Public channel	Server S
U_i selects ID_i, PW_i, random number r_i		
Computes $PWR_i = h(ID_i \parallel r_i \parallel PW_i)$	$\{ID_i, PWR_i\}$ $\Longrightarrow$ smart card $\Longleftarrow$	Computes $A_i = h(ID_i \parallel x)$ $B_i = A_i \oplus PWR_i$ $DID_i = E_x(ID_i \parallel N)$
Stores r_i in smart card		Stores $\{B_i, DID_i, E_{Key}(\cdot)/D_{Key}(\cdot)\}$ in a smart card
Inputs ID_i^*, PW_i^*		
Chooses a random number RN_i	$\{DID_i, M_i, T_i\}$ $\longrightarrow$	Computes $(ID_i \parallel N) = D_x(DID_i)$
Computes $A_i = B_i \oplus h(ID_i^* \parallel r_i \parallel PW_i^*)$		$A_i' = h(ID_i \parallel x)$
$M_i = E_{A_i}(ID_i^* \parallel RN_i \parallel T_i \parallel DID_i)$		$(ID_i' \parallel RN_i' \parallel T_i' \parallel DID_i') = D_{A_i'}(M_i)$
		Checks $DID_i' \stackrel{?}{=} DID_i$, $T_i' \stackrel{?}{=} T_i$
		Chooses random numbers N^{new}, RN_S
Computes $\left(DID_i^{new} \parallel RN_S{}' \parallel ID_i'' \parallel RN_i'\right) = D_{A_i}(P_i)$	$\{P_i\}$ $\longleftarrow$	Computes $DID_i^{new} = E_x(ID_i \parallel N^{new})$
Checks $RN_i' \stackrel{?}{=} RN_i$, $ID_i'' \stackrel{?}{=} ID_i$		$P_i = E_{A_i'}\left(DID_i^{new} \parallel RN_S \parallel ID_i \parallel RN_i'\right)$
Computes $Q_i = h(RN_S{}' \parallel DID_i^{new} \parallel RN_i)$	$\{Q_i\}$ $\longrightarrow$	
$SK = h(RN_S{}' \parallel A_i \parallel RN_i)$		Computes $Q_i' = h(RN_S \parallel DID_i^{new} \parallel RN_i{}')$
		Checks $Q_i' \stackrel{?}{=} Q_i$
		Computes $SK = h(RN_S \parallel A_i' \parallel RN_i{}')$

Figure 2: Nikooghadam et al.'s scheme.

Step 3. Upon receiving $\{L_a, Y_i, T_a\}$, as T_a is fresh, S computes ${L_a}^d \bmod m$ to retrieve $(ID_i' \parallel D_a' \parallel N_{1_a}')$. Then S computes $A_i' = h(d \parallel ID_i')$, $[h(PWR_i^* \parallel ID_i)]' = Y_i \oplus A_i'$, $D_a'' = h(A_i' \parallel [h(PWR_i^* \parallel ID_i)]' \parallel T_a \parallel N_{1_a}')$. As D_a'' is equal to D_a', S regards $\mathscr{A}$ as legitimate user U_i.

The inherent reason for above attacks is that there is a verification value B_i in smart card for the adversary to check if the guessed ID_i and PW_i are correct.

2.2.3. Forward Secrecy. Suppose $\mathscr{A}$ gets the secret key $\{d, u, v\}$ and intercepts message $\{L_i, Y_i, T_u\}$ and $\{X_i, Z_i, T_s\}$ from public channel. Then $\mathscr{A}$ calculates the session key as follows.

Step 1. Compute ${L_i}^d \bmod m$ to derive $(ID_i' \parallel D_i' \parallel N_1')$.

Step 2. Compute $N_2' = Z_i \oplus N_1'$.

Step 3. Compute $A_i' = h(d \parallel ID_i')$, $SK = h(N_1' \parallel A_i' \parallel N_2')$.

In Amin et al.'s scheme, U_i and S select random numbers N_1, N_2, respectively. The transmission of N_1 uses RSA encryption under the public key of S. The transmission of N_2 uses bitwise XOR with N_1. Hence, the confidentiality of random numbers is completely dependent on the private key of S. Once the private key of S is compromised, the attacker can easily get all session keys of the whole server system.

3. Cryptanalysis of Nikooghadam Et Al.'s Scheme

In this section, we review Nikooghadam et al.'s scheme and reveal its security vulnerabilities.

3.1. Review of Nikooghadam Et Al.'s Scheme. Nikooghadam et al.'s scheme includes the following two phases (see Figure 2).

3.1.1. Registration Phase. In this phase, when receiving the enrollment request, S issues a smart card to U_i.

Step 1. U_i picks ID_i, PW_i freely. Then U_i calculates $PWR_i = h(ID_i \parallel r_i \parallel PW_i)$, where r_i is a random number. $\{ID_i, PWR_i\}$ is transmitted to S through a secure channel.

Step 2. After receiving $\{ID_i, PWR_i\}$, S calculates $A_i = h(ID_i \parallel x)$, $B_i = A_i \oplus PWR_i$. S computes U_i's dynamic identity $DID_i = E_x(ID_i \parallel N)$, where N is a random number. S stores $\{B_i, DID_i, E_{Key}(\cdot)/D_{Key}(\cdot)\}$ in a smart card and delivers it to U_i through a secure channel.

Step 3. U_i stores r_i in the smart card.

3.1.2. Login and Authentication Phase. This phase verifies the legitimacy of communicating parties and negotiates a session key.

Step 1. U_i inserts his smart card into a terminal and enters ID_i^*,PW_i^*. The smart card calculates $A_i = B_i \oplus h(ID_i^* \parallel r_i \parallel PW_i^*)$, $M_i = E_{A_i}(ID_i^* \parallel RN_i \parallel T_i \parallel DID_i)$, where RN_i is a nonce. The smart card delivers $\{DID_i, M_i, T_i\}$ to S.

Step 2. After receiving $\{DID_i, M_i, T_i\}$, S checks the freshness of T_i. Then S computes $(ID_i \parallel N) = D_x(DID_i)$, $A_i' = h(ID_i \parallel x)$, $(ID_i' \parallel RN_i' \parallel T_i' \parallel DID_i') = D_{A_i'}(M_i)$, and checks $DID_i' \stackrel{?}{=} DID_i$, $T_i' \stackrel{?}{=} T_i$. If both the two equations hold, perform next step. Otherwise, this protocol aborts.

Step 3. S chooses two random numbers N^{new}, RN_S. Then S computes $DID_i^{new} = E_x(ID_i \parallel N^{new})$, $P_i = E_{A_i'}(DID_i^{new} \parallel RN_S \parallel ID_i \parallel RN_i')$, and returns $\{P_i\}$ to U_i.

Step 4. Upon receiving $\{P_i\}$, the smart card calculates $(DID_i^{new} \parallel RN_S{}' \parallel ID_i'' \parallel RN_i{}') = D_{A_i}(P_i)$ and checks if $RN_i{}' = RN_i, ID_i'' = ID_i$. If the two equations hold, perform next step; otherwise, this protocol aborts.

Step 5. The smart card calculates $Q_i = h(RN_S{}' \parallel DID_i^{new} \parallel RN_i)$, $SK = h(RN_S{}' \parallel A_i \parallel RN_i)$. $\{Q_i\}$ is transmitted to S.

Step 6. Upon receiving $\{Q_i\}$, S calculates $Q_i' = h(RN_S \parallel DID_i^{new} \parallel RN_i{}')$ and checks $Q_i' \stackrel{?}{=} Q_i$. If the equation holds, S computes $SK = h(RN_S \parallel A_i' \parallel RN_i{}')$.

3.2. Cryptanalysis of Nikooghadam Et Al.'s Scheme. In this part, we demonstrate Nikooghadam et al.'s scheme is susceptible to several security attacks.

3.2.1. Inefficiency for Wrong Password Detection. In the login phase, the smart card never checks the validity of the entered password. If U_i inputs a wrong password by accident, the smart card cannot detect this fault, until a login request is sent to S, and S returns back a response to reject it. It wastes too much time of users.

3.2.2. Offline Password Guessing Attack. The adversary $\mathscr{A}$ extracts $\{B_i, DID_i, E_{Key}(\cdot)/D_{Key}(\cdot), r_i\}$ from U_i's smart card and intercepts $\{DID_i, M_i, T_i\}$ from public channel. Offline password guessing is launched in the following steps.

Step 1. Choose an identity ID_i^* from the identity dictionary space and a password PW^* from the password dictionary space.

Step 2. Calculate $A_i^* = B_i \oplus h(ID^* \parallel r_i \parallel PW^*)$, $(ID_i^{**} \parallel RN_i^* \parallel T_i^* \parallel DID_i^*) = D_{A_i^*}(M_i)$. Check $DID_i^* \stackrel{?}{=} DID_i$, $T_i^* \stackrel{?}{=} T_i$. If both the two equations hold, it shows that ID_i^* is U_i's real identity, and PW^* is U_i's correct password.

Step 3. Repeat Steps 1-2, until $\mathscr{A}$ find real ID_i and PW_i.

3.2.3. User Anonymity. For the message $\{DID_i, M_i, T_i\}$, ID_i is compromised by performing offline password guessing attack; this is violation of user anonymity.

Once the adversary $\mathscr{A}$ gets $\{B_i, DID_i, E_{Key}(\cdot)/D_{Key}(\cdot), r_i\}$ and obtains ID_i and PW_i by performing offline password guessing attack and intercepts $\{DID_i, M_i, T_i\}$, $\{P_i\}$, and $\{Q_i\}$ from pubic channel, $\mathscr{A}$ performs user impersonation attack, server impersonation attack, man-in-the-middle attack, session key disclosure attack, and desynchronization attack as follows.

3.2.4. User Impersonation Attack.

Step 1. $\mathscr{A}$ computes $A_i = B_i \oplus h(ID_i \parallel r_i \parallel PW_i)$.

Step 2. $\mathscr{A}$ computes $M_a = E_{A_i}(ID_i \parallel RN_{i_a} \parallel T_a \parallel DID_i)$, where RN_{i_a} is a nonce, and T_a is current timestamp. $\mathscr{A}$ sends $\{DID_i, M_a, T_a\}$ to S.

Step 3. After S receiving $\{DID_i, M_a, T_a\}$, as T_a is fresh, $DID_i' = DID_i$, $T_a' = T_a$, S regards $\mathscr{A}$ as the legitimate user U_i and returns $\{P_a\}$ to $\mathscr{A}$.

Step 4. Upon receiving $\{P_a\}$, $\mathscr{A}$ computes $(DID_i^{new} \parallel RN_S{}' \parallel ID_i'' \parallel RN_{i_a}{}') = D_{A_i}(P_a)$, $Q_a = h(RN_S{}' \parallel DID_i^{new} \parallel RN_{i_a})$, $SK = h(RN_S{}' \parallel A_i \parallel RN_{i_a})$ and sends $\{Q_a\}$ to S.

Step 5. After S receiving $\{Q_a\}$, as $Q_a' = Q_a$, S computes $SK = h(RN_S \parallel A_i' \parallel RN_{i_a}{}')$.

$\mathscr{A}$ establishes a session key SK with S successfully.

3.2.5. Server Impersonation Attack.

Step 1. Compute $A_i = B_i \oplus h(ID_i \parallel r_i \parallel PW_i)$.

Step 2. Compute $(ID_i' \parallel RN_i' \parallel T_i' \parallel DID_i') = D_{A_i}(M_i)$.

Step 3. Intercept $\{P_i\}$ from public channel. Generate a nonce RN_{S_a} and select a binary string $DID_i{}^a$ whose length is the same as DID_i. Compute $P_a = E_{A_i}(DID_i{}^a \parallel RN_{S_a} \parallel ID_i \parallel RN_i')$ and return $\{P_a\}$ to U_i.

Step 4. After receiving $\{P_a\}$, as $RN_i{}' = RN_i, ID_i'' = ID_i, U_i$ regards $\mathscr{A}$ as the sever S.

3.2.6. Man-in-the-Middle Attack.

Step 1. Intercept $\{DID_i, M_i, T_i\}$ from public channel and send $\{DID_i, M_a, T_a\}$ to S.

Step 2. Intercept $\{P_i\}$ from public channel and send $\{P_a\}$ to U_i.

Step 3. Intercept $\{Q_i\}$ from public channel and send $\{Q_a\}$ to S.

3.2.7. Session Key Disclosure Attack.

Step 1. Calculate $A_i = B_i \oplus h(ID_i \parallel r_i \parallel PW_i)$.

Step 2. Calculate $(DID_i^{new} \parallel RN_S{}' \parallel ID_i'' \parallel RN_i{}') = D_{A_i}(P_i)$.

Step 3. Calculate $SK = h(RN_S{}' \parallel A_i \parallel RN_i{}')$.

3.2.8. Desynchronization Attack.

Step 1. Calculate $A_i = B_i \oplus h(ID_i \parallel r_i \parallel PW_i)$.

Step 2. Intercept $\{P_i\}$ from pubic channel. Compute $(DID_i^{New} \parallel RN_S{}' \parallel ID_i'' \parallel RN_i') = D_{A_i}(P_i)$. Select a binary string $DID_i{}^a$ whose length is same as DID_i^{New}. Compute $P_a = E_{A_i}(DID_i{}^a \parallel RN_S{}' \parallel ID_i \parallel RN_i')$. Sends $\{P_a\}$ to U_i.

Step 3. Intercept $\{Q_i\}$ from pubic channel. Compute $Q_a = h(RN_S' \| DID_i^{new} \| RN_i')$. Send Q_a to S.

After that, U_i cannot login S anymore, unless U_i re-register to S.

3.2.9. Forward Secrecy. Suppose $\mathscr{A}$ obtains the master key of S and intercepts $\{DID_i, M_i, T_i\}$ and $\{P_i\}$ from public channel, and then $\mathscr{A}$ calculates the session key as follows.

Step 1. Calculate $ID_i = D_x(DID_i)$.

Step 2. Calculate $A_i = h(ID_i \| x)$.

Step 3. Calculate $(DID_i^{New} \| RN_S' \| ID_i'' \| RN_i') = D_{A_i}(P_i)$.

Step 4. Calculate $SK = h(RN_S' \| A_i \| RN_i')$.

The transmission of random numbers RN_i, RN_s uses symmetric encryption under key A_i. The confidentiality of random numbers is dependent on the authentication value A_i. Once the master key of S is leaked, the attacker is able to compute $A_i = h(ID_i \| x)$ and obtain RN_i, RN_s by decrypting message $\{P_i\}$. Consequently, the session keys of the whole sever system are compromised.

In the authentication scheme, random numbers are essential to establish unique session key in each session. If the user or server cannot transmit random number to the other side securely, it certainly will compromise the session key.

4. Cryptanalysis of Two Authentication Schemes for Specific Environment

Recently, Mishra et al. presented an efficient authentication scheme for session initiation protocol. In addition, Hsieh et al. introduced an authentication scheme for wireless communication. However, after a rigorous analysis, we discover both the two schemes have vulnerabilities. We reveal the weaknesses of the two schemes in this section.

4.1. Review of Mishra Et Al.'s Scheme. We briefly review Mishra et al.'s scheme in this subsection. As the password and biometric update phase is irrelevant to our cryptanalysis, we omit it.

4.1.1. Registration Phase.

Step 1. U_i picks ID_i, PW_i at will and computes $RPW_i = h(ID_i \| PW_i \| r)$, where r is a nonce. U_i delivers $\{ID_i, RPW_i\}$ to the server S securely.

Step 2. Upon receiving $\{ID_i, RPW_i\}$, S computes $X_i = h(mk \| ID_i \| N)$, $Y_i = X_i \oplus RPW_i$, where mk is the secret key of S, and N is the number of times the user once registered with S. S sends a smart card storing $\{Y_i, mkP\}$ to U_i.

Step 3. U_i imprints his biometric B_i. The smart card computes $V = h(ID_i \| PW_i \| h(B_i))$, $R = h(B_i) \oplus r$, and stores V, R in its memory.

4.1.2. Login and Authentication Phase.

Step 1. U_i enters ID_i^*, PW_i^* and imprints B_i^*. The smart card checks if $V = h(ID_i^* \| PW_i^* \| h(B_i^*))$. If it holds, the smart card calculates $r = h(B_i^*) \oplus R$, $RPW_i = h(ID_i^* \| PW_i^* \| r)$, $X_i = Y_i \oplus RPW_i$, $D_1 = h(ID_i^* \| X_i \| h((u \bullet mkP)_x) \| (uP)_x \| T_1)$, $DID_i = h((u \bullet mkP)_x) \oplus ID_i^*$, where u is a nonce. The smart card delivers $\{DID_i, D_1, uP, T_1\}$ to S.

Step 2. After receiving the message, S computes $ID_i = h((mk \bullet uP)_x) \oplus DID_i$, $X_i = h(mk \| ID_i \| N)$, $D_1' = h(ID_i \| X_i \| h((mk \bullet uP)_x \| (uP)_x \| T_1)$ and checks if $D_1' = D_1$. If it holds, S calculates $sk = h(ID_i \| X_i \| (mk \bullet uP)_x) \| T_1 \| T_3)$, $D_2 = h(sk \| T_1 \| T_3 \| (mk \bullet uP)_x)$ and sends $\{D_2, T_3\}$ to U_i.

Step 3. After receiving $\{D_2, T_3\}$, the smart card calculates $sk = h(ID_i \| X_i \| (u \bullet mkP)_x \| T_1 \| T_3)$, $D_2' = h(sk \| T_1 \| T_3 \| (u \bullet mkP)_x)$. If $D_2' = D_2$, U_i regards he establishes a valid session key with S.

4.2. Weaknesses of Mishra Et Al.'s Scheme

4.2.1. Forward Secrecy. In the case that the adversary $\mathscr{A}$ obtains the secret key mk and intercepts the messages $\{DID_i, D_1, uP, T_1\}$ and $\{D_2, T_3\}$ from public channel, then $\mathscr{A}$ is able to breach the session key in the following steps.

Step 1. Compute $ID_i = DID_i \oplus h((mk \bullet uP)_x)$.

Step 2. Let $N = 0$.

Step 3. Compute $X_i = h(mk \| ID_i \| N)$, $D_1' = h(ID_i \| X_i \| h((mk \bullet uP)_x)) \| (uP)_x \| T_1)$.

Step 4. Check if $D_1' = D_1$. If they are equal, proceed next step. Otherwise, let $N = N + 1$, and go to Step 3.

Step 5. Compute the session key $sk = h(ID_i \| X_i \| (mk \bullet uP)_x) \| T_1 \| T_3)$.

4.3. Review of Hsieh Et Al.'s Scheme. We briefly review Hsieh et al.'s scheme in this subsection. As the ticket authentication phase is irrelevant to our cryptanalysis, we omit it.

4.3.1. Registration Phase.

Step 1. The mobile station MS delivers $\{ID_i, PW_i\}$ to the home agent HA securely.

Step 2. After receiving the registration request, HA computes $z_3 = z_1 \oplus y$, $A = h(ID_i \| x) \oplus y \oplus z_2$, $D_i = ID_i \oplus h(ID_{HA} \| x) \oplus z_1$, $z_4 = PW_i \oplus y \oplus z_2$, where x, y are the secret values of HA, and z_1, z_2 are two random numbers. HA issues a smart card storing $\{ID_{HA}, z_3, A, D_i, z_4\}$ to MS.

4.3.2. Ticket-Issuing Phase.

Step 1. MS computes $R_1 = aP$, $z_5 = z_3 \oplus N_1$, $h(ID_i \| x) = PW_i \oplus z_4 \oplus A$, $B = h(ID_i \| x) \oplus R_1$, $DID_i = D_i \oplus N_1$, where

a, N_1 are random numbers. *MS* sends $\{ID_{HA}, z_5, B, DID_i\}$ to the foreign agent *FA*.

Step 2. After receiving the message, *FA* computes $R_2 = bP$, $C = E_k(ID_{FA}, T_1, R_2)$, where k is a secret key shared by *FA* and *HA*. *FA* delivers $\{ID_{FA}, z_5, B, DID_i, C\}$ to *HA*.

Step 3. After receiving the message, *HA* computes $N_1 = z_5 \oplus z_1 \oplus y$, $ID_i = h(ID_{HA} \parallel x) \oplus z_1 \oplus N_1$, $R_1 = h(ID_i \parallel x) \oplus B$, $(ID_{FA}, T_1, R_2) = D_k(C)$, $R_3 = cR_1$, $R_4 = cR_2$, $M_1 = h(h(ID_i \parallel x) \parallel R_1) \oplus R_4$, $M_2 = h(h(ID_i \parallel x) \parallel R_1 \parallel R_4)$, $M_3 = h(h(ID_i \parallel x) \parallel R_1 + 1 \parallel R_4 + 1)$, $F = E_k(T_3, R_3, M_3)$. *HA* delivers $\{M_1, M_2, F\}$ to *FA*.

Step 4. Upon receiving $\{M_1, M_2, F\}$, *FA* computes $(T_3, R_3, M_3) = D_k(F)$, $SK = bR_3$, $t = E_{SK}(ticket)$. *FA* sends $\{M_1, M_2, t\}$ to *MS*.

Step 5. Upon receiving $\{M_1, M_2, t\}$, *MS* computes $R_4 = h(h(ID_i \parallel x) \parallel R_1) \oplus M_1$, $M_2' = h(h(ID_i \parallel x) \parallel R_1 \parallel R_4)$ and checks if $M_2' = M_2$. If it holds, *MS* calculates $SK = aR_4$, $ticket = D_{SK}(t)$, $M_3' = h(h(ID_i \parallel x) \parallel R_1 + 1 \parallel R_4 + 1)$, and sends $\{M_3'\}$ to *FA*.

Step 6. Upon receiving $\{M_3'\}$, *FA* checks if $M_3' = M_3$. If the equation holds, *FA* and *MS* authenticate each other and establish a session key successfully.

4.4. Weaknesses of Hsieh Et Al.'s Scheme

4.4.1. Offline Password Guessing Attack. Suppose the adversary $\mathcal{A}$ extracts $\{ID_{HA}, z_3, A, D_i, z_4\}$ from the smart card and intercepts the messages $\{ID_{HA}, z_5, B, DID_i\}$ and $\{M_1, M_2, t\}$ from public channel. Then $\mathcal{A}$ performs the following steps.

Step 1. $\mathcal{A}$ picks a password PW^* from the password dictionary space.

Step 2. $\mathcal{A}$ computes $h(ID_i \parallel x)^* = PW^* \oplus z_4 \oplus A$, ${R_1}^* = h(ID_i \parallel x)^* \oplus B$, ${R_4}^* = h(h(ID_i \parallel x)^* \parallel {R_1}^*) \oplus M_1$, $M_2^* = h(h(ID_i \parallel x)^* \parallel {R_1}^* \parallel {R_4}^*)$ and checks if $M_2^* = M_2$. If it holds, it shows that PW^* is the correct password.

Step 3. Repeat Steps 1-2, until $\mathcal{A}$ finds PW_i.

4.4.2. MS Impersonation Attack. In the case that smart card is compromised, the adversary is able to obtain the password via offline password guessing attack; that is to say, the adversary has acquired all the authentication information that *MS* has. The capability of adversary has no differences with the legitimate *MS*. Hence, the adversary is able to impersonate *MS* and defraud *FA* and *HA* successfully.

4.4.3. HA Impersonation Attack. $\mathcal{A}$ intercepts the message $\{ID_{HA}, z_5, B, DID_i\}$ from public channel. With the smart card and PW_i, $\mathcal{A}$ is able to perform *HA* impersonation attack in the following steps.

Step 1. $\mathcal{A}$ computes $h(ID_i \parallel x) = PW_i \oplus z_4 \oplus A$, $R_1 = h(ID_i \parallel x) \oplus B$, $R_4 = dP$, $M_1 = h(h(ID_i \parallel x) \parallel R_1) \oplus R_4$, $M_2 = h(h(ID_i \parallel x) \parallel R_1 \parallel R_4)$, $SK = dR_1$, $t = E_{SK}(ticket)$, where d is a random number. $\mathcal{A}$ sends $\{M_1, M_2, t\}$ to *MS*.

Step 2. After receiving $\{M_1, M_2, t\}$, *MS* computes $R_4 = h(h(ID_i \parallel x) \parallel R_1) \oplus M_1$, $M_2' = h(h(ID_i \parallel x) \parallel R_1 \parallel R_4)$. As $M_2' = M_2$, *MS* computes $SK = aR_4$, $ticket = D_{SK}(t)$. *MS* believes that it establishes a session key with *HA*.

5. Proposed Scheme

To overcome the weaknesses of Amin et al.'s scheme and Nikooghadam et al.'s scheme, we propose an improved anonymous authentication protocol using ECC. The proposed scheme establishes secure session key based on Diffie-Hellman key exchange. Our scheme consists of the following four phases (see Figure 3).

5.1. Initialization Phase. S chooses an elliptic curve group E_p. P is a generator of E_p. Then, S chooses a random number x as its private key and calculates $P_{pub} = xP$ as its public key. S publishes $\{E_p, P, P_{pub}\}$ and keeps x as secret.

5.2. Registration Phase. In this phase, when receiving a registration request, S issues a smart card containing the parameters for user authentication to U_i.

Step 1. U_i picks ID_i and PW_i freely. U_i calculates $P_i = h(PW_i \oplus r_i)$, where r_i is a random number. Afterwards, $\{ID_i, P_i\}$ is transmitted to S via a secure channel.

Step 2. After receiving $\{ID_i, P_i\}$, S calculates $A_i = h(x \parallel ID_i)$, $B_i = A_i \oplus P_i$, $V_i = h(P_i \parallel ID_i) \bmod n$, where $2^4 \le n \ll 2^8$. S stores $\{B_i, V_i, E_{Key}(\cdot), E_p, P, P_{pub}, n\}$ in a smart card and transmits it to U_i through a secure communication channel.

Step 3. U_i stores r_i in the smart card.

5.3. Login and Authentication Phase. This phase verifies the authenticity of communicating parties and generates a session key.

Step 1. U_i attaches his smart card to a terminal and inputs ID_i^* and PW_i^*. Then the smart card calculates $P_i^* = h(PW_i^* \oplus r_i)$, $V_i^* = h(P_i^* \parallel ID_i^*) \bmod n$, and checks $V_i^* \stackrel{?}{=} V_i$. If the equation holds, the smart card computes $R_i = N_1 P$, $C_i = h(N_1 P_{PUB})$, $A_i^* = B_i \oplus P_i^*$, $D_i = h(A_i^*) \oplus R_i$, $L_i = E_{C_i}(ID_i^* \parallel D_i)$, where N_1 is a random number. $\{L_i, R_i\}$ is transmitted to S.

Step 2. After receiving $\{L_i, R_i\}$, S computes $C_i' = h(xR_i)$, $(ID_i' \parallel D_i') = D_{C_i'}(L_i)$, $A_i' = h(x \parallel ID_i')$, $D_i'' = h(A_i') \oplus R_i$, and checks $D_i'' \stackrel{?}{=} D_i'$. If the equation holds, S generates a random number N_2 and computes $Z_i = N_2 P$, $K_i = N_2 \bullet R_i$, $SK = h(K_i \parallel A_i')$, $X_i = h(Z_i \parallel D_i'' \parallel SK)$. $\{Z_i, X_i\}$ is transmitted to U_i.

Step 3. After receiving $\{Z_i, X_i\}$, U_i computes $K_i' = N_1 \bullet Z_i$, $SK = h(K_i' \parallel A_i^*)$, $X_i' = h(Z_i \parallel D_i \parallel SK)$, and checks $X_i' \stackrel{?}{=} X_i$. If the equation holds, S computes $E_i = h(D_i \parallel K_i' \parallel SK)$. $\{E_i\}$ is transmitted to S.

U_i/Smart card	Channel	Server S
U_i selects ID_i, PW_i, random number r_i	$\{ID_i, P_i\}$ $\Longrightarrow$	
Computes $P_i = h(PW_i \oplus r_i)$		Computes $A_i = h(ID_i \parallel x)$
		$B_i = A_i \oplus P_i$
	smart card $\Longleftarrow$	$V_i = h(P_i \parallel ID_i) \bmod n$
Stores r_i into smart card		Stores $\{B_i, V_i, E_{Key}(\cdot), E_p, P, P_{pub}, n\}$ in a smart card
Inputs ID_i^*, PW_i^*		
Computes $P_i^* = h(PW_i^* \oplus r_i)$	$\{L_i, R_i\}$ $\longrightarrow$	Computes $C_i' = h(xR_i)$, $(ID_i' \parallel D_i') = D_{C_i'}(L_i)$
$V_i^* = h(P_i^* \parallel ID_i^*) \bmod n$		
Checks $V_i^* \stackrel{?}{=} V_i$		Calculates $A_i' = h(x \parallel ID_i')$
Generates a nonce N_1		$D_i'' = h(A_i') \oplus R_i$
Computes $R_i = N_1 P$		Checks $D_i'' \stackrel{?}{=} D_i'$
$C_i = h(N_1 P_{PUB})$		Chooses a random number N_2
$A_i^* = B_i \oplus P_i^*$	$\{Z_i, X_i\}$ $\longleftarrow$	Computes $Z_i = N_2 P$,
Computes $D_i = h(A_i^*) \oplus R_i$		$K_i = N_2 \bullet R_i$
$L_i = E_{C_i}(ID_i^* \parallel D_i)$		$SK = h(K_i \parallel A_i')$
		$X_i = h(Z_i \parallel D_i'' \parallel SK)$
Computes $K_i' = N_1 \bullet Z_i$		
$SK = h(K_i' \parallel A_i^*)$	$\{E_i\}$ $\longrightarrow$	
$X_i' = h(Z_i \parallel D_i \parallel SK)$		
Checks $X_i' \stackrel{?}{=} X_i$		
Computes $E_i = h(D_i \parallel K_i' \parallel SK)$		Computes $E_i' = h(D_i'' \parallel K_i \parallel SK)$
		Checks $E_i' \stackrel{?}{=} E_i$

Figure 3: The proposed scheme.

Step 4. After receiving $\{E_i\}$, S computes $E_i' = h(D_i'' \parallel K_i \parallel SK)$ and checks $E_i' \stackrel{?}{=} E_i$. If the equation holds, S establishes a session key with U_i successfully.

5.4. Password Updates Phase. U_i updates his original password to a new one as follows.

Step 1. U_i attaches his smart card to a terminal and inputs ID_i^* and PW_i^*. The smart card calculates $P_i^* = h(PW_i^* \oplus r_i)$, $V_i^* = h(P_i^* \parallel ID_i^*) \bmod n$, and checks $V_i^* \stackrel{?}{=} V_i$. If it holds, the smart card asks the user to input a new password.

Step 2. U_i enters his new password PW_i^{new}. Then the smart card calculates $B_i^{new} = B_i \oplus P_i^* \oplus h(PW_i^{new} \oplus r_i)$, $V_i^{new} = h(h(PW_i^{new} \oplus r_i) \parallel ID_i^*) \bmod n$. The smart card deletes B_i, V_i and stores B_i^{new}, V_i^{new} in its memory.

6. Security Analysis

6.1. Informal Analysis. The heuristic analysis shows that our scheme can withstand various known attacks.

6.1.1. Offline Password Guessing Attack. Suppose that the adversary $\mathcal{A}$ steals the smart card of U_i and extracts $\{B_i, V_i, E_{Key}(\cdot), E_p, P, P_{pub}, n, r_i\}$ from it; then $\mathcal{A}$ executes the following steps.

Step 1. Choose an identity ID_i^* from identity dictionary space and a password PW_i^* from password dictionary space.

Step 2. Compute $P_i^* = h(PW_i^* \oplus r_i)$, $V_i^* = h(P_i^* \parallel ID_i^*) \bmod n$.

Step 3. Check $V_i^* \stackrel{?}{=} V_i$.

In our scheme, we employ the verification value $V_i^* = h(P_i^* \parallel ID_i^*) \bmod n$. When $n = 2^8$ and the identity and password are 64 bits, there are $(2^{64} * 2^{64})/2^8$ pairs of $< ID_i^*, PW_i^* >$ conforming to $V_i^* = V_i$. Even if $\mathcal{A}$ finds a pair of $< ID_i^*, PW_i^* >$ that conforms to $V_i^* = V_i$, the probability they are equal to the identity and password of U_i is $2^8/(2^{64} * 2^{64})$.

Our scheme overcomes the offline password guessing attack at a cost of false acceptance rate of $1/2^8$. But it does not compromise the security. When the server receives a login request message generated with erroneous $< ID_i^*, PW_i^* >$, as $D_i'' \neq D_i'$, the server rejects the login request.

6.1.2. Replay Attack. Suppose that the adversary $\mathcal{A}$ tries to launch replay attack in the following cases.

In the case that the adversary $\mathcal{A}$ intercepts $\{L_i, R_i\}$ from public channel and replays the message to S, S handles the message and returns $\{Z_i, X_i\}$ to $\mathcal{A}$. However, without N_1 and A_i, $\mathcal{A}$ is unable to generate a valid $\{E_i\}$. The protocol finally aborts.

In the case that $\mathcal{A}$ replays message $\{Z_i, X_i\}$ or $\{E_i\}$. Both the two messages are generated based on the random numbers N_1 and N_2. The random numbers are only valid for the current session. If $\mathcal{A}$ replays $\{Z_i, X_i\}$, the smart card will find that the received X_i is not equal with X_i' computed

based on its secret. Similarly, if $\mathscr{A}$ replays $\{E_i\}$, S will find that $E_i \neq E_i'$. The protocol finally aborts.

6.1.3. Desynchronization Attack. Desynchronization attack denotes that the adversary modifies the parameters of a message or simply blocks the message causing that the legitimate user cannot access the server anymore.

In our scheme, all the messages are generated based on the authentication value A_i and the random number N_1 or N_2. The user and the server check the validity of each message they receive. If the adversary $\mathscr{A}$ modifies the parameters of a message, the receiver will detect the message is tampered with and reject it. Besides, if $\mathscr{A}$ blocks a message, it just leads to the single authentication failure for the current session, but it does not alter the parameters that the user and the server have. The user is able to continue to access the server.

6.1.4. User Anonymity. In our scheme, ID_i is protected by symmetric encryption under the key C_i. The adversary $\mathscr{A}$ cannot retrieve ID_i, unless he gets the private key of S. Furthermore, the cipher text of ID_i changes with random number N_1 in each session. Our scheme also achieves user identity untraceability.

6.1.5. User Impersonation Attack. $\mathscr{A}$ extracts $\{B_i, V_i, E_{Key}(\cdot), E_p, P, P_{pub}, n, r_i\}$ from the stolen smart card and intercepts the transmitted messages between U_i and S from public channel. Then $\mathscr{A}$ tries to forge a login request $\{L_i, R_i\}$ to defraud S. $\mathscr{A}$ generates a random number N_a and computes $R_i = N_1 P$, $C_i = N_1 P_{PUB}$. To compute L_i, A_i is required. However, A_i is protected by hash function. The only way to obtain A_i is to retrieve A_i from B_i. As $A_i^* = B_i \oplus P_i^*$, $P_i^* = h(PW_i^* \oplus r_i)$, the adversary needs to get PW_i firstly. But as analyzed in "offline password guessing attack", the adversary is unable to get PW_i in the case that smart card is compromised. Eventually, the adversary cannot forge a valid login request. Our scheme is immune to user impersonation attack.

6.1.6. Server Impersonation Attack. Provided that $\mathscr{A}$ gets $\{B_i, V_i, E_{Key}(\cdot), E_p, P, P_{pub}, n, r_i\}$ and the transmitted messages between U_i and S,$\mathscr{A}$ tries to impersonate the server by forging a valid message $\{Z_i, X_i\}$, where $Z_i = N_2 P$, $X_i = h(Z_i \parallel D_i \parallel SK)$, $D_i = h(A_i) \oplus R_i$, $SK = h(K_i \parallel A_i)$. In order to compute X_i, the adversary has to know A_i. As analyzed in "user impersonation attack", the adversary is unable to obtain A_i, even if he compromises the smart card. Without A_i, the adversary is unable to perform server impersonation attack successfully.

6.1.7. Forward Secrecy. Assuming the adversary $\mathscr{A}$ obtains the private key of S and intercepts $\{L_i, R_i, T_i\}$, $\{X_i, Z_i\}$, $\{E_i\}$ from public channel. Then $\mathscr{A}$ tries to compute the session key. As $SK = h(K_i \parallel A_i)$, K_i and A_i are required to compute SK. With the private key x, the adversary computes $C_i' = xR_i$, $(ID_i' \parallel D_i') = D_{C_i'}(L_i)$, $A_i = h(ID_i \parallel x)$. A_i is known. Next, to obtain K_i, $\mathscr{A}$ needs to derive $K_i(N_1 N_2 P)$ from $R_i(N_1 P)$, $Z_i(N_2 P)$. That is to say, $\mathscr{A}$ needs to solve the elliptic curve Diffie–Hellman problem (CDHP). Otherwise, the adversary $\mathscr{A}$ is unable to reveal SK. Hence, our scheme achieves forward secrecy.

Table 2: The notations and rules of BAN logic.

P, Q	A principal
X	A statement
$P\mid \equiv X$	P believes X is true
$P \lhd X$	P see X, P receives a message that includes X
$P\mid \sim X$	P said X, P once sent a message containing X
$P \Longrightarrow X$	P has jurisdiction over X
$\#(X)$	X is fresh
$P \overset{K}{\longleftrightarrow} Q$	P and Q have a shared key K
$< X >_K$	X is combined with a secret K
Message meaning rule	$\dfrac{P\mid \equiv P \overset{K}{\longleftrightarrow} Q, P \lhd < X >_K}{P\mid \equiv Q\mid \sim X}$
nonce-verification rule	$\dfrac{P\mid \equiv \#(X), P\mid \equiv Q\mid \sim X}{P\mid \equiv Q\mid \equiv X}$
jurisdiction rule	$\dfrac{P\mid \equiv Q \Longrightarrow X, P\mid \equiv Q\mid \equiv X}{P\mid \equiv X}$

6.1.8. Session Key Disclosure Attack. $\mathscr{A}$ intercepts the transmitted messages between U_i and S from public channel. Then $\mathscr{A}$ attempts to compromise SK. A_i and K_i are required to compute SK. To get A_i, $\mathscr{A}$ needs to break the smart card and password of user at the same time. To get K_i, $\mathscr{A}$ needs to solve the elliptic curve CDHP. Both are beyond the ability of $\mathscr{A}$. Our scheme is secure against session key disclosure attack.

In our scheme, the session key includes a long-term authentication value and a temporary secret key generated by Diffie-Hellman key exchange. The long-term authentication value denotes the shared secret authentication information between S and U_i. The temporary secret key ensures that our scheme is resistant to known key attack and achieves forward secrecy.

6.2. Formal Analysis. We use BAN logic [38] to prove our scheme achieves mutual authentication and session key establishment. Table 2 gives the symbols and rules used in BAN logic.

The goals that our scheme should achieve are as follows.

Goal 1: $U_i\mid \equiv S\mid \equiv (S \overset{SK}{\longleftrightarrow} U_i)$

Goal 2: $U_i\mid \equiv (S \overset{SK}{\longleftrightarrow} U_i)$

Goal 3: $S\mid \equiv U_i\mid \equiv (S \overset{SK}{\longleftrightarrow} U_i)$

Goal 4: $S\mid \equiv (S \overset{SK}{\longleftrightarrow} U_i)$

The idealized form of our scheme is given as follows.

M1: $U_i \longrightarrow S < N_1 P, S \overset{C_i}{\longleftrightarrow} U_i >_{A_i}$

Table 3: Results of security analysis.

Schemes	Nikooghadam [15]	Luo [17]	Xie [19]	Amin [20]	Maitra [22]	Maitra [23]	Islam [24]	Our protocol
S1	×	×	√	×	×	×	×	√
S2	×	√	√	√	√	√	×	√
S3	×	×	√	×	×	×	√	√
S4	×	×	√	√	×	×	√	√
S5	×	×	√	√	×	×	√	√
S6	×	√	√	√	√	√	√	√
S7	×	√	√	×	×	√	√	√
S8	×	×	√	√	√	√	√	√
S9	×	×	×	√	√	√	√	√
S10	√	√	√	√	×	√	√	√

S1: Resist offline password guessing attack. S2: User anonymity. S3: Resist user impersonation attack. S4: Resist server impersonation attack. S5: Resist man-in-the-middle attack. S6: Resist desynchronization attack. S7: Forward secrecy. S8: Resist session key disclosure attack. S9: Efficient wrong password detection. S10: Resist insider attack.

M2: $S \longrightarrow U_i < N_2P, S \overset{SK}{\longleftrightarrow} U_i >_{A_i}$

M3: $U_i \longrightarrow S < N_2 \bullet R_i, S \overset{SK}{\longleftrightarrow} U_i >_{A_i}$

The assumptions about the initial belief of our scheme are given as follows.

S1: $S| \equiv (S \overset{A_i}{\longleftrightarrow} U_i)$

S2: $S| \equiv \#(N_1)$

S3: $S| \equiv U_i \Longrightarrow (S \overset{C_i}{\longleftrightarrow} U_i)$

S4: $U_i| \equiv (S \overset{A_i}{\longleftrightarrow} U_i)$

S5: $U_i| \equiv \#(N_1)$

S6: $U_i| \equiv S \Longrightarrow (S \overset{SK}{\longleftrightarrow} U_i)$

S7: $S| \equiv \#(N_2)$

S8: $S| \equiv U_i \Longrightarrow (S \overset{SK}{\longleftrightarrow} U_i)$

The BAN logic proof of our scheme is performed in the following steps.

According to M1, we have

(1) $S \lhd < N_1P, S \overset{C_i}{\longleftrightarrow} U_i >_{A_i}$

Applying the message-meaning rule on S1, (1), we obtain

(2) $S| \equiv U_i| \sim (N_1P, S \overset{C_i}{\longleftrightarrow} U_i)$

Applying the nonce-verification rule on S2, (2), we obtain

(3) $S| \equiv U_i| \equiv (S \overset{C_i}{\longleftrightarrow} U_i)$

Applying the jurisdiction rule on S3, (3), we obtain

(4) $S| \equiv (S \overset{C_i}{\longleftrightarrow} U_i)$

According to M2, we have

(5) $U_i \lhd < N_2P, S \overset{SK}{\longleftrightarrow} U_i >_{A_i}$

Applying the message-meaning rule on S4, (5), we obtain

(6) $U_i| \equiv S| \sim < N_2P, S \overset{SK}{\longleftrightarrow} U_i >_{A_i}$

Applying the nonce-verification rule on S5, (6), we obtain

(7) $U_i| \equiv S| \equiv (S \overset{SK}{\longleftrightarrow} U_i))$ *Goal 1*

Applying the jurisdiction rule to S6, (7), we obtain

(8) $U_i| \equiv (S \overset{SK}{\longleftrightarrow} U_i)$ *Goal 2*

According to M3, we have

(9) $S \lhd < N_2 \bullet R_i, S \overset{SK}{\longleftrightarrow} U_i >_{A_i}$

Applying the message-meaning rule on S1, (9), we obtain

(10) $S| \equiv U_i| \sim (N_2 \bullet R_i, S \overset{SK}{\longleftrightarrow} U_i)$

Applying the nonce-verification rule on S7, (10), we obtain

(11) $S| \equiv U_i| \equiv (S \overset{SK}{\longleftrightarrow} U_i)$ *Goal 3*

Applying the jurisdiction rule to S8, (11), we obtain

(12) $S| \equiv (S \overset{SK}{\longleftrightarrow} U_i)$ *Goal 4*

7. Security and Performance Comparison

We compare the proposed scheme with other recently introduced schemes [15, 17, 19, 20, 22–24] in this section. Table 3 shows the results of security analysis. Only our scheme is resistant to various known attacks and achieves forward secrecy and user anonymity and detects the wrong password promptly, while other schemes have security loopholes more or less.

We estimate the computation overheads and communication costs of related schemes concerning login and authentication phase in Table 4. In accordance with Table 4, the total computation overhead of our scheme is inferior to

Table 4: Performance comparison of our protocol with other schemes.

Schemes ↓	User computation cost	Server computation cost	Estimated total computation time	Communication overhead
Nikooghadam et al. [15]	$2T_{SE} + 3T_H$	$4T_{SE} + 3T_H$	0.0414 ms	1152 bits
Luo et al. [17]	$4T_{PM} + 4T_H$	$4T_{PM} + 5T_H$	17.8287 ms	960 bits
Xie et al.[19]	$3T_{PM} + 6T_H$	$3T_{PM} + 2T_{SE} + 5T_H$	13.3905 ms	1088 bits
Amin et al. [20]	$1T_{ME} + 6T_H$	$1T_{ME} + 4T_H$	7.723 ms	1664 bits
Maitra et al. [22]	$6T_{ME} + 8T_H$	$4T_{ME} + 12T_H$	38.546 ms	4480 bits
Maitra et al. [23]	$3T_{PM} + 5T_H$	$4T_{PM} + 3T_H$	15.6004 ms	1248 bits
Islam [24]	$5T_{ME} + 5T_H$	$2T_{ME} + 4T_H$	26.9707 ms	2688 bits
Our protocol	$2T_{PM} + 1T_{SE} + 7T_H$	$3T_{PM} + 1T_{SE} + 6T_H$	11.1691 ms	704 bits

T_{ME} denotes one modular exponentiation. T_{PM} denotes one point multiplication on elliptic curve group. T_{SE} denotes one symmetric encryption/decryption. T_H denotes performing a hash function. According to [37], the computing times of T_H, T_{SE}, T_{PM}, T_{ME} are 0.0023 ms, 0.0046 ms, 2.226 ms, and 3.85 ms respectively. The computing time of lightweight operation "XOR" is negligible. To compute the communication overhead, we suppose that ID_i and random number and timestamp are 64 bits, the output of hash function and symmetric encryption are 128 bits, large-prime numbers m, p are 1024 bits, one point on the elliptic curve E_p is 160 bits.

Amin et al.'s scheme and Nikooghadam et al.'s scheme, but it is pretty better than other schemes [17, 19, 22–24]. Furthermore, we have demonstrated that the schemes of Amin et al. and Nikooghadam et al. suffer from various vulnerabilities. In addition, our scheme has the lowest communication overhead among these schemes.

In summary, our scheme has obvious advantages in terms of security. As for computation overhead, our scheme is superior to most schemes. Besides, our scheme is ahead of other schemes in aspect of communication cost. Hence, our scheme is more practical than other schemes.

8. Conclusion

In this paper, we cryptanalyze two anonymous authentication schemes and pointed out they suffer from a variety of security attacks like offline password guessing attack and user impersonation attack. In addition, we reveal the security weaknesses of two environment-specific authentication schemes. To overcome the security loopholes, we present an efficient authentication scheme using elliptic curve cryptosystem. In our scheme, the security of session key is fully guaranteed. It prevents all the possible session key exposure, forward secrecy attack included. We give both formal analysis and informal analysis to prove the completeness and security of the proposed scheme. Furthermore, the outcomes of security and performance comparison show that the proposed scheme is superior with better security and low computation and communication cost. Besides, the proposed scheme has good expansibility. In the future, we plan to design an efficient biometrics-based remote user authentication scheme for multiserver environment based on our current work.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by the National Key Research and Development Program of China No. 2018YFB0803605 and the National Natural Science Foundation of China No. 61897069.

References

[1] T. Kumar, A. Braeken, M. Liyanage, and M. Ylianttila, "Identity privacy preserving biometric based authentication scheme for Naked healthcare environment," in *Proceedings of the IEEE International Conference on Communications, (ICC '17)*, Paris, France, 2017.

[2] F. Wei, J. Ma, Q. Jiang, J. Shen, and C. Ma, "Cryptanalysis and improvement of an enhanced two-factor user authentication scheme in wireless sensor networks," *Information Technology & Control*, vol. 45, no. 1, pp. 62–70, 2016.

[3] L. Wu, J. Fan, Y. Xie, J. Wang, and Q. Liu, "Efficient location-based conditional privacy-preserving authentication scheme for vehicle ad hoc networks," *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, 2017.

[4] M. Khan and S. Kim, "Cryptanalysis and security enhancement of a more efficient and secure dynamic ID-based remote user authentication scheme," *Computer Communications*, vol. 34, no. 3, pp. 305–309, 2011.

[5] Y.-F. Chang, W.-L. Tai, and H.-C. Chang, "Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 3430–3440, 2014.

[6] R.-C. Wang, W.-S. Juang, and C.-L. Lei, "Robust authentication and key agreement scheme preserving the privacy of secret key," *Computer Communications*, vol. 34, no. 3, pp. 274–280, 2011.

[7] F. Wen and X. Li, "An improved dynamic ID-based remote user authentication with key agreement scheme," *Computers and Electrical Engineering*, vol. 38, no. 2, pp. 381–387, 2012.

[8] S. Wu, Y. Zhu, and Q. Pu, "Robust smart-cards-based user authentication scheme with user anonymity," *Security and Communication Networks*, vol. 5, no. 2, pp. 236–248, 2012.

[9] D. Wang, P. Wang, C. Ma, and Z. Chen, "Robust smart card based password authentication scheme against smart card security breach," Cryptology ePrint Archive Report 2012/439, 2012, https://eprint.iacr.org/2012/439.pdf.

[10] R. Madhusudhan and R. C. Mittal, "Dynamic ID-based remote user password authentication schemes using smart cards: a review," *Journal of Network and Computer Applications*, vol. 35, no. 4, pp. 1235–1248, 2012.

[11] K. Kim and M. Kim, "An enhanced anonymous authentication and key exchange scheme using smart card," in *Proceedings of the International Conference on Information Security and Cryptology*, pp. 487–494, Springer-Verlag, 2012.

[12] S. Islam and G. Biswas, "Dynamic ID-based remote user mutual authentication scheme with smart card using elliptic curve cryptography," *Journal of Electronics*, vol. 31, no. 5, pp. 473–488, 2014.

[13] H.-F. Huang, H.-W. Chang, and P.-K. Yu, "Enhancement of timestamp-based user authentication scheme with smart card," *International Journal of Network Security*, vol. 16, no. 6, pp. 463–467, 2014.

[14] D. Wang, N. Wang, P. Wang, and S. Qing, "Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity," *Information Sciences*, vol. 321, Article ID 11496, pp. 162–178, 2015.

[15] M. Nikooghadam, R. Jahantigh, and H. Arshad, "A lightweight authentication and key agreement scheme preserving user anonymity," *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 13401–13423, 2017.

[16] J. Jung, J. Kim, Y. Choi, and D. Won, "An anonymous user authentication and key agreement scheme based on a symmetric cryptosystem in wireless sensor networks," *Sensors*, vol. 16, no. 8, article 1299, 2016.

[17] M. Luo, Y. Zhang, M. K. Khan, and D. He, "A secure and efficient identity-based mutual authentication scheme with smart card using elliptic curve cryptography," *International Journal of Communication Systems*, vol. 30, no. 16, p. e3333, 2017.

[18] L. Xiong, D. Peng, T. Peng, H. Liang, and Z. Liu, "A lightweight anonymous authentication scheme with perfect forward secrecy for wireless sensor networks," *Sensors*, vol. 17, no. 11, 2016.

[19] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, "Provably secure dynamic ID-based anonymous two-factor authenticated key exchange scheme with extended security model," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1382–1392, 2017.

[20] R. Amin, T. Maitra, D. Giri, and P. D. Srivastava, "Cryptanalysis and improvement of an RSA based remote user authentication scheme using smart card," *Wireless Personal Communications*, vol. 96, no. 3, pp. 4629–4659, 2017.

[21] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang, "Targeted online password guessing: An underestimated threat," in *Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS '16)*, pp. 1242–1254, Vienna, Austria, 2016.

[22] T. Maitra, M. S. Obaidat, R. Amin, S. H. Islam, S. A. Chaudhry, and D. Giri, "A robust ElGamal-based password-authentication scheme using smart card for client-server communication," *International Journal of Communication Systems*, vol. 30, no. 11, 2016.

[23] T. Maitra, M. S. Obaidat, S. H. Islam, D. Giri, and R. Amin, "Security analysis and design of an efficient ECC based two factor password authentication scheme," *Security and Communication Networks*, vol. 9, no. 17, pp. 4166–4181, 2016.

[24] S. H. Islam, "Design and analysis of an improved smartcard-based remote user password authentication scheme," *International Journal of Communication Systems*, vol. 29, no. 11, pp. 1708–1719, 2016.

[25] P. Chandrakar and H. Om, "A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC," *Computer Communications*, vol. 110, pp. 26–34, 2017.

[26] O. Mir, J. Munilla, and S. Kumari, "Efficient anonymous authentication with key agreement scheme for wireless medical sensor networks," *Peer-to-Peer Networking and Applications*, vol. 10, no. 1, pp. 79–91, 2017.

[27] D. He, S. Zeadally, N. Kumar, and W. Wu, "Efficient and anonymous mobile user authentication scheme using self-certified public key cryptography for multi-server architectures," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2052–2064, 2016.

[28] S. Kumari, M. K. Khan, X. Li, and F. Wu, "Design of a user anonymous password authentication scheme without smart card," *International Journal of Communication Systems*, vol. 29, no. 3, pp. 441–458, 2016.

[29] K.-H. Yeh, "A lightweight authentication scheme with user untraceability," *Frontiers of Information Technology and Electronic Engineering*, vol. 16, no. 4, pp. 259–271, 2015.

[30] D. Kang, J. Jung, J. Mun, D. Lee, Y. Choi, and D. Won, "Efficient and robust user authentication scheme that achieve user anonymity with a Markov chain," *Security and Communication Networks*, vol. 9, no. 11, pp. 1462–1476, 2016.

[31] B. Djellali, K. Belarbi, A. Chouarfia, and P. Lorenz, "User authentication scheme preserving anonymity for ubiquitous devices," *Security and Communication Networks*, vol. 8, no. 17, pp. 3131–3141, 2015.

[32] Y. Lu, L. Li, H. Peng, and Y. Yang, "Robust anonymous two-factor authenticated key exchange scheme for mobile client-server environment," *Security and Communication Networks*, vol. 9, no. 11, pp. 1331–1339, 2016.

[33] D. Mishra, A. K. Das, and S. Mukhopadhyay, "A secure and efficient ECC-based user anonymity-preserving session initiation authentication protocol using smart card," *Peer-to-Peer Networking and Applications*, vol. 9, no. 1, pp. 171–192, 2016.

[34] W.-B. Hsieh and J.-S. Leu, "Anonymous authentication protocol based on elliptic curve Diffie-Hellman for wireless access networks," *Wireless Communications and Mobile Computing*, vol. 14, no. 10, 2014.

[35] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2016.

[36] S. Halevi and H. Krawczyk, "Public-key cryptography and password protocols," *ACM Transactions on Information and System Security*, vol. 2, no. 3, pp. 230–268, 1998.

[37] H. Kilinc and T. Yanik, "A survey of SIP authentication and key agreement schemes," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1005–1023, 2014.

[38] M. Burrows, M. Abadi, and M. Needham, "Logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.