Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 3048697, 13 pages
https://doi.org/10.1155/2018/3048697

Research Article

An Enhanced User Authentication Protocol Based on Elliptic Curve Cryptosystem in Cloud Computing Environment

Chenyu Wang[1], **Ke Ding**[2], **Bin Li**[3], **Yiming Zhao**[3], **Guoai Xu**[1], **Yanhui Guo**[1] and **Ping Wang**[2,3]

[1]*School of Cyberspace Security, Beijing University of Posts and Telecommunications, China*
[2]*School of EECS, Peking University, China*
[3]*School of Software and Microelectronics, Peking University, China*

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

Received 13 April 2018; Revised 24 June 2018; Accepted 12 July 2018; Published 1 October 2018

Academic Editor: Jian Shen

With the popularity of cloud computing, information security issues in the cloud environment are becoming more and more prominent. As the first line of defense to ensure cloud computing security, user authentication has attracted extensive attention. Though considerable efforts have been paid for a secure and practical authentication scheme in cloud computing environment, most attempts ended in failure. The design of a secure and efficient user authentication scheme for cloud computing remains a challenge on the one hand and user's smart card or mobile devices are of limited resource; on the other hand, with the combination of cloud computing and the Internet of Things, applications in cloud environments often need to meet various security requirements and are vulnerable to more attacks. In 2018, Amin et al. proposed an enhanced user authentication scheme in cloud computing, hoping to overcome the identified security flaws of two previous schemes. However, after a scrutinization of their scheme, we revealed that it still suffers from the same attacks (such as no user anonymity, no forward secrecy, and being vulnerable to offline dictionary attack) as the two schemes they compromised. Consequently, we take the scheme of Amin et al. (2018) as a study case, we discussed the inherent reason and the corresponding solutions to authentication schemes for cloud computing environment in detail. Next, we not only proposed an enhanced secure and efficient scheme, but also explained the design rationales for a secure cloud environment protocol. Finally, we applied BAN logic and heuristic analysis to show the security of the protocol and compared our scheme with related schemes. The results manifest the superiority of our scheme.

1. Introduction

With the development of IT technology, cloud computing has become one of the hottest research directions in recent years. As a new type of service, cloud computing is rapidly integrated into our daily lives with its high scalability, high service efficiency, and low-cost charge [1]. It fundamentally changed the traditional model of service providers providing services and consumers' access to resources: as a service provider (such as Google, Microsoft, Amazon) of cloud computing, it effectively improves the utilization of resources by centralizing the demands; consumers not only gain the convenience of using resources, but also reduce the using cost through paying on demand. Therefore, more and more big firms build their own cloud platforms and provide services, including Google App Engine, Amazon Web services, and IBM SmartCloud [2]. Furthermore, both the individuals and small companies enjoy the benefits of cloud services. Generally speaking, there are three kinds of cloud services: (1) Iaas, Infrastructure as a Service, which means providing user with the infrastructure such as storage and networks to use; (2) Paas, Platform as a Service, which means providing user with the platform to develop various applications; (3) Saas, Software as a Service, which means providing user with software applications [3, 4].

However, with the increasing popularity of the cloud services, the security issues have become more prominent, how to protect user privacy and restrict data from being

illegally accessed has become a challenging problem and research hotspot. The first step to solve these issues is user authentication which can verify the authenticity of communication participants. A secure user authentication scheme will firstly verify the authenticity of the user when he/she applies to access the cloud data; then to prevent a malicious cloud server trick users, the validity of the cloud server should be checked; once confirming the identity of the user and the cloud server, a session key will be established to encrypt the communication messages.

Generally speaking, there are three ways to authenticate a user, which are based on the following: (1) what you know (such as the password); (2) what you have (such as the smart card); (3) who you are (such as the biometric characteristic: fingerprint and iris). Due to its simplicity and practicality, passwords have been used more widely. While a password-based authentication protocol has natural flaws, it cannot resist against offline dictionary guessing attacks. Consequently, as a factor to help enhance security, the smart card gets used [5–9]. A scheme combined two factors (such as the password and the smart card) is called two-factor user authentication scheme. The participants of the two-factor authentication scheme in cloud computing environment involves a user, a cloud server and a register authority. Note that among the three participants, only the register authority is trusted. At first, the user and the cloud server register to the register authority, respectively. Then the cloud server will send the user a smart card with some sensitive information and negotiate a shared secret parameter with the cloud server. Later on, when the user initiates an access request to the cloud server in login phase, the three participants will authenticate themselves to each other. If they all are authenticated, the user will be allowed to access the cloud server.

Motivations. In 2013, Yang et al. [10] devoted to design a secure authentication scheme for cloud computing environment, while their scheme is vulnerable to dictionary attack. Then Yang et al. [11] proposed a new scheme; unfortunately, Chen et al. [12] then showed this scheme is not secure to insider attack and impersonation attack and proposed a new version which once again is broken by Wang et al. [13]. Wang et al. [13] pointed out that Chen et al.'s scheme [12] is subject to offline dictionary attack and impersonation attack. Most recently, Amin et al. [3] identified the security weaknesses in the schemes of Xue et al. [14] and Chuang et al. [15] by revealing the two schemes fail to provide user anonymity and forward secrecy while being not able to resist against offline password guessing attack and so on. Therefore, they designed a new scheme that claims to overcome the security flaws of the two schemes and be secure to various attacks. However, after a scrutinization of Amin et al.'s scheme, we found their scheme still cannot overcome their identified security threats.

In these years, considerable efforts have been paid for a secure and practical authentication scheme in cloud computing environment, some typical schemes including [16–19], yet most of them are found having security flaws more or less. Designing of a secure authentication scheme for cloud computing environment is still a challenge. With the widespread use of cloud computing, the potential security threats will lead to greater harm. This unsatisfactory situation motivates us to explore the inherent reasons of the failure in those schemes, find the basic method to fix the security flaws, and design a robust and efficient user authentication protocol for cloud computing environment.

Our contributions. Amin et al.'s scheme [3] is a very typical scheme which suffers from the common attacks, while the scheme's structure is widely accepted. So we take Amin et al.'s protocol as a study case to elaborate the common issues (and its corresponding solutions) in most authentication schemes and provide rationales for designing a secure cloud environment protocol. In addition, based on the analysis, we design a secure authentication protocol. In a short, our contributions can be summarized as follows:

(1) We demonstrated that Amin et al.'s scheme [3] fails to achieve user anonymity and forward secrecy while being not able to resist against offline dictionary attack.

(2) We discussed the inherent reasons of the identified flaws and its corresponding solutions; furthermore, we realized the way of deploying a public key algorithm rightly is challenging. Therefore, we showed the essential points for deploying public key algorithms.

(3) We improved Amin et al.'s scheme from security and effectiveness two aspects, proved the security of our scheme via BAN logic and heuristic analysis and, finally, compared our scheme with other related schemes. The results show that our scheme is more suitable for cloud computing environment.

The remainder of this paper is organized as follows: Section 2 sketches complexity assumptions and extends adversary model; then, Amin et al.'s scheme is reviewed and analyzed in Section 3; in Section 4, we propose a secure scheme and elaborate on design rationales; Section 5 proves the security of our scheme; in Section 6, we compare our scheme with other related schemes; finally, the conclusion is drawn in Section 7.

2. Preliminary

This section introduces the preliminary of the whole paper, including complexity assumptions in designing a scheme and some notations and abbreviations.

2.1. Computational Problems. Given two large primes p and q, let $\mathbb{F}_p$ be a finite field, $E/\mathbb{F}_p$ be an elliptic curve over $\mathbb{F}_p$, and $\mathbb{G}$ be a q-order subgroup of $E/\mathbb{F}_p$. For $\alpha, \beta \in Z_p^*$ and a point P in $\mathbb{G}$, we can define the discrete logarithm problem as follows:

(1) Elliptic curve discrete logarithm (ECDL) problem: given $(P, \alpha P)$, it is impossible to compute α within polynomial time.

(2) Elliptic curve computational Diffie-Hellman (ECCDH) problem: given $(\alpha P, \beta P)$, it is impossible to compute $\alpha\beta P$ within polynomial time.

2.2. Adversary Models. Understanding the adversary models is the most basic step to design and analyze a protocol. In

Table 1: Notations and abbreviations.

Symbol	Description
U_i	i^{th} user
CS_j	j^{th} cloud server
RA	the register authority
$\mathscr{A}$	the adversary
x	the long term secret key of RA
y	the secret key of RA
ID_i	identity of U_i
PW_i	password of U_i
SID_j	identity of CS_j
$Skey_j$	shared key between G_k and CS_j
$\oplus$	bitwise XOR operation
$\|$	concatenation operation
$h(\cdot)$	one-way hash function
$\longrightarrow$	a common channel
$\Longrightarrow$	a secure channel

2015, Wang et al. [20] proposed the capabilities of adversary in distributed systems: (1) exhaust passwords and identities; (2) learn ID_i when evaluating security strength; (3) control of the open communication channel; (4) learn PW_i or extract information in the smart card; (5) acquire previous session keys; (6) know the long-term secret key x when considering forward secrecy. As both the distributed systems and the cloud computing systems have similar network environment, their adversary models are also similar too. Therefore, we adopt Wang et al.'s adversary models [20] which have been accepted by various schemes [21–23].

2.3. Notations and Abbreviations. As shown in Table 1, we summarize the notations and abbreviations used in this paper.

3. Cryptanalysis of Amin et al.'s Scheme

After identifying the security pitfalls in other two user authentication schemes, Amin et al. [3] attempted to design a new light weight protocol in cloud computing environment. After analyzing their scheme using AVISPA tool, they claimed the new scheme achieves forward security while being resistant to various attacks. However, this section will show that, under the assumptions on adversary capabilities in Section 2.2, their scheme cannot provide forward security while being subject to two kinds of offline dictionary attacks [27] and so on. Thus their scheme is not a truly two-factor scheme. To address these issues, this section first reviews the scheme of Amin et al. and then analyzes Amin et al.'s scheme [3].

3.1. Review of Amin et al.'s Scheme. This section briefly reviews the scheme of Amin et al. [3]; their scheme consists of five phases. As the password change phase and identity update phase have little relevance, we omit them. Furthermore, we adjust some symbols of their scheme for the ease of reading and the unity of the paper.

Registration Phase

(1) Cloud Server Registration Phase

Step 1. $CS_j \Longrightarrow RA$: $\{SID_j, d\}$. d is a random number.

Step 2. $RA \Longrightarrow CS_j$: $\{Skey_j\}$. RA computes $PSID_j = h(SID_j \| d)$, $Skey_j = h(PSID_j \| y)$.

Step 3. CS_j keeps $\{Skey_j, d\}$.

(2) User Registration Phase

Step 1. $U_i \Longrightarrow RA$: $\{A_i, PID_i\}$. $A_i = h(PW_i \| b_1)$, $PID_i = h(ID_i \| b_2)$, $bb_i = b_2 \oplus A_i$ where b_1, b_2 are two random number chosen by U_i.

Step 2. $RA \Longrightarrow U_i$: smart card $\{C_i, E_i, bb_i, DP, h(\cdot)\}$. RA calculates $C_i = h(A_i \| PID_i)$, $D_i = h(PID_i \| x)$, $E_i = D_i \oplus A_i$.

Step 3. U_i inputs DP, bb_i into the card where $DP = h(ID_i \| PW_i) \oplus b_1$.

Login and Authentication Phase

Step 1. $U_i \longrightarrow CS_j$: $\{G_i, F_i, Z_i, PID_i, TS_i\}$. After U_i inputs ID_i and PW_i, the card computes $b_1^* = DP \oplus h(ID_i^* \| PW_i^*)$, $A_i^* = h(PW_i^* \| b_1^*)$, $b_2^* = bb_i \oplus A_i^*$, $PID_i^* = h(ID_i^* \| b_2^*)$, $C_i^* = h(A_i^* \| PID_i^*)$. If $(C_i^* == C_i)$, the card produces a random number N_i (128 bit) and computes $D_i = E_i \oplus A_i$, $G_i = h(PID_i \| SID_j \| N_i \| TS_i \| D_i)$, $F_i = D_i \oplus N_i$, $Z_i = SID_j \oplus h(D_i \| N_i)$.

Step 2. $CS_j \longrightarrow RA$: $\{J_i, K_i, PSID_j, G_i, F_i, Z_i, PID_i, TS_i, TS_j\}$. CS_j first checks $(TS_j - TS_i < \Delta T)$, then selects a 128 bit random number N_j, and computes $J_i = Skey_j \oplus N_j$, $K_i = h(N_j \| Skey_j \| G_i \| TS_j)$.

Step 3. $RA \longrightarrow CS_j$: $\{P_{cs}, R_{cs}, Q_{cs}, V_{cs}\}$. RA checks $(TS_{cs} - TS_j < \Delta T^*)$ and computes $D_i = h(PID_i \parallel x)$, $N_i^* = F_i \oplus D_i$, $SID_j^* = Z_i \oplus h(D_i \parallel N_i^*)$, $G_i^* = h(PID_i \parallel SID_j^* \parallel N_i^* \parallel D_i \parallel TS_i)$.

If $(G_i^* == G_i)$, compute $Skey_j^* = h(PSID_j \parallel y)$, $N_j^* = Skey_j^* \oplus J_i$, $K_i^* = h(Skey_j^* \parallel N_j^* \parallel G_i \parallel TS_j)$.

If $(K_i^* == K_i)$, compute $P_{cs} = N_j \oplus N_{cs} \oplus h(N_i \parallel D_i)$, $R_{cs} = N_i \oplus N_{cs} \oplus h(Skey_j^* \parallel N_j^*)$, $SK_{cs} = h(N_i \oplus N_j \oplus N_{cs})$, $Q_{cs} = h((N_j \oplus N_{cs}) \parallel SK_{cs})$, $V_{cs} = h((N_i \parallel N_{cs}) \parallel SK_{cs})$

Step 4. $CS_j \longrightarrow U_i$: $\{P_{cs}, Q_{cs}\}$. CS_j computes $W_j = h(Skey_j \parallel N_j)$, $N_i \oplus N_{cs} = R_{cs} \oplus W_j$, $SK_j = h(N_i \oplus N_{cs} \oplus N_j)$, $V_{cs}^* = h((N_i \oplus N_{cs}) \parallel SK_j)$. If $(V_{cs}^* \neq V_{cs})$, send $\langle P_{cs}, Q_{cs} \rangle$ to U_i.

Step 5. U_i computes $L_i = h(N_i \parallel D_i)$, $N_j \oplus N_{cs} = P_{cs} \oplus L_i$, $SK_i = h(N_j \oplus N_{cs} \oplus N_i)$, $Q_{cs}^* = h((N_j \oplus N_{cs}) \parallel SK_i)$. If $(Q_{cs}^* == Q_{cs})$, the whole authentication phase finishes successfully.

3.2. Cryptanalysis of Amin et al.'s Scheme. Amin et al.'s scheme [3] does not point out the adversarial model, while, according to their attack on the scheme of Xue et al. [14] and Chuang et al. [15], we can infer their adversary model which is included into our model (see Section 2.2). Although Amin et al.'s scheme [3] provides many admirable features, such as changing password locally and high efficiency, it still suffers from various attacks like most authentication protocol in cloud computing environment. Therefore, their scheme is a typical case to show the security threat in cloud environment. Through Amin et al.'s scheme, we can get insight into the inherent reasons of the failure in other authentication protocols for cloud and, based on it, learn to design a secure one. In brief, this section, on one hand, demonstrates that Amin et al.'s scheme [3] is vulnerable to various attacks and, on the other hand, indicates the failure reasons of their scheme.

Off-Line Dictionary Attack I. ***(i) The adversary's capability***: it obtains the message $\{C_i, E_i, bb_i, DP\}$ in U_i's smart card.

(ii) The attack steps:the steps are as follows.

Step 1. Guess PW_i to be PW_i^*, ID_i to be ID_i^*. Note that, it is quite realistic for $\mathcal{A}$ to obtain the password and identity simultaneously, because their spaces are limited [28].

Step 2. Compute $b_1^* = DP \oplus h(ID_i^* \parallel PW_i^*)$.

Step 3. Compute $A_i^* = h(PW_i^* \parallel b_1^*)$.

Step 4. Compute $b_2^* = bb_i \oplus A_i^*$.

Step 5. Compute $PID_i^* = h(ID_i^* \parallel b_2^*)$.

Step 6. Compute $C_i^* = h(A_i^* \parallel PID_i^*)$.

Step 7. Verify the correctness of PW_i and ID_i by checking if $(C_i^* \stackrel{?}{=} C_i)$.

Step 8. Repeat steps 1 ~ 6 until the correct values of PW_i and ID_i are found.

(iii) The time complexity: $\mathcal{O}(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * 4T_H)$, where T_H is the time of hash-function.

Remark. Generally speaking, achieving two-factor security is the most essential requirement of a two-factor authentication protocol; that is, any one of the factors being broken will not trigger the security of another factor, which in turn threatens the entire system. In recent years, many protocols have tried to propose a secure two-factor security protocol, but most have failed. It was not until the work of Ma et al. [29] and Wang et al. [20, 30] did such a stagnant situation completely changed. In 2012, Ma et al. [29] pointed out that public key algorithm is necessary to design a secure two-factor authentication scheme; in 2015, Wang et al. [20] found that there is a conflict between changing password locally and resisting against smart card loss attack under the current technique; therefore, Wang et al. [30] put forward a way of "honeywords"+"fuzzy-verifier" to solve the conflict; in 2016, Wang et al. [27] further pointed out that there are two offline dictionary attacks and then combined with the results of [29, 30] and matched the corresponding solutions for each attack.

In this paper, we follow the classification method of Wang et al. [27] and demonstrate that Amin et al.'s scheme [3] cannot resist against the two kinds of dictionary attack. Looking back at the above attack process, we can find that the key to the problem is that $\mathcal{A}$ can find the verification value C_i to check the correctness of the guessed result. According to Wang et al. [30], this issue can be settled with the integration of "fuzzy-verifier" and "honeywords": let $C_i = h(A_i \parallel PID_i \parallel D_i) \bmod n_0$ where n_0 is a integer $(2^4 \leq n_0 \leq 2^8)$. The detailed explanation on this method can be found in Section IV of [30] or Section 5.2 of this paper.

Off-Line Dictionary Attack II. ***(i) The adversary's capability***: (1) it eavesdrops on one of U_i's login requests $\{G_i, F_i, Z_i, PID_i, TS_i\}$ and (2) obtains the message $\{C_i, E_i, bb_i, DP\}$ in U_i's smart card.

(ii) The attack steps:the steps are as follows.

Step 1. Guess PW_i to be PW_i^*, ID_i to be ID_i^*.

Step 2. Compute $b_1^* = DP \oplus h(ID_i^* \parallel PW_i^*)$.

Step 3. Compute $A_i^* = h(PW_i^* \parallel b_1^*)$.

Step 4. Compute $D_i^* = E_i \oplus A_i^*$.

Step 5. Compute $N_i^* = F_i \oplus D_i^*$.

Step 6. Compute $G_i^* = h(PID_i \parallel SID_j \parallel N_i^* \parallel TS_i \parallel D_i^*)$. Note that Amin et al. [3] view SID_j as a secret only known to the legitimate user. However, it not practical: $\mathcal{A}$ at least can register as a legitimate user to get SID_j.

Step 7. Verify the correctness of PW_i and ID_i by checking if $(G_i^* == G_i)$.

Step 8. Repeat steps 1 ~ 6 until the correct values of PW_i and ID_i are found.

(iii) The time complexity: $\mathcal{O}(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * 3T_H)$.

Remark. In this attack, the pivotal parameter is G_i. To adversary $\mathcal{A}$, the only challenge in computing G_i is the value of D_i which can be derived from (ID_i, PW_i), so once $\mathcal{A}$ guesses the value of (ID_i, PW_i), he/she can check the correctness of them via G_i. Now considering a situation where G_i consists of the secret shared D_i and an another nonpublic dynamic parameter which should not be derived from (ID_i, PW_i). In this situation, $\mathcal{A}$ cannot use G_i to check the guessed value anymore, since there is another uncertain parameter besides D_i. Consequently, constructing such a dynamic parameter which is known to U_i and RA is our critical step to address this attack. Taking into account the fact that Ma et al.'s emphasizes [29] on the necessity of lightweight public-key algorithm in designing a secure authentication protocol, we then apply a lightweight public-key algorithm to construct such a dynamic parameter. In addition, our specific ideas on solving this attack are shown in Section 4.

User anonymity: these days user anonymity has become one of the security issues that people are widely concerned about, especially in the case of cloud computing and the Internet of Things that involve massive data. The adversary can acquire people's sensitive personal information via various ways including analyzing the session transcript in the open channel when the services are accessed [31]. Moreover, with the development of the technology, the adversary may even trace users' movement and learn the location of their home or company, which triggers a huge potential threat [32]. Under these circumstances, user anonymity is a pivotal attribute of the authentication scheme to protect user privacy.

Generally speaking, user anonymity covers two aspects [31]: (1) user identity protection; (2) user untraceability. The former requires the scheme does not expose users' identity; and the latter prevents an adversary from linking the session transcripts to a specific user or distinguishes the sessions sent by different users. This definition on user anonymity is widely applied in most authentication schemes [8, 24, 33, 34]. Unfortunately, in Amin et al.'s scheme, the parameter PID_i that identifies the user identity is a static value exposed in the insure channel, which means the adversary can trace U_i via PID_i. Consequently, Amin et al.'s scheme cannot provide user anonymity. As we can see, one of the keys to achieve user anonymity is concealing the real identity with a dynamic parameter. The way to implement this is called dynamic-ID technique [20]. According to Wang et al.'s suggestion [31], we can employ the dynamic-ID technique to protect user anonymity via applying a lightweight public-key algorithm to the authentication schemes as described in Section 4.

Forward Secrecy. ***(i) The adversary's capability***: (1) it eavesdrops on $\{F_i, PID_i\}$ and $\{P_{cs}\}$ and (2) obtains the long-term key x.

(ii) The attack steps:the steps are as follows.

Step 1. Compute $D_i^* = h(PID_i \parallel x)$.

Step 2. Compute $N_i^* = F_i \oplus D_i^*$.

Step 3. Compute $L_i^* = h(N_i^* \parallel D_i^*)$.

Step 4. Compute $(N_j \oplus N_{cs})^* = L_i^* \oplus P_{cs}$.

Step 5. Compute $SK = h((N_j \oplus N_{cs})^*) \oplus N_i^*$.

(iii) The time complexity: $O(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * 3T_H)$.

Remark. In Amin et al.' scheme [3], there are two secret keys (x, y) in the register authority. We have shown that the leakage of the long-term key x will lead to the exposure of previous sessions key. In the following attack, we can see that the leakage of y leading to the same question too. As a result, the use of two system parameters is of little significance but consumes resource.

(i) The adversary's capability: (1) it eavesdrops on $\{J_i, PSID_j\}$ and $\{R_{cs}\}$ and (2) obtains the secret key y.

(ii) The attack steps:the steps are as follows.

Step 1. Compute $Skey_j^* = h(PID_j \parallel y)$.

Step 2. Compute $N_j^* = J_i \oplus Skey_j^*$.

Step 3. Compute $W_j^* = h(Skey_j^* \parallel N_j)$.

Step 4. Compute $(N_i \oplus N_{cs})^* = W_j^* \oplus R_{cs}$.

Step 5. Compute $SK = h(N_j \oplus (N_i \oplus N_{cs})^*)$.

(iii) The time complexity: $\mathcal{O}(|\mathcal{D}_{pw}| * |\mathcal{D}_{id}| * 3T_H)$.

Remark. When considering the forward secrecy, the adversary almost has the same capacity with RA except that $\mathcal{A}$ does not know the verifier-table. As a result, if RA can compute the session key according to the processes of the scheme, then $\mathcal{A}$ is very likely to break the session key. For the above considerations, we do not recommend that RA have the ability to calculate session keys. To achieve this, a public-key algorithm is suggested too [29]. In addition, the more concrete improved methods will be explained in Section 4.

Other flaws: using timestamps to resist replay attack is not recommended. As we all know, due to the network congestion, network latency, or other issues, maintaining a consistent network clock between different systems is very difficult, which often results in the desynchronization attacks. As a matter of fact, many papers [20, 30] in their evaluation criteria pointed out that a protocol using timestamps cannot resist against desynchronization attacks. Furthermore, determining an appropriate value of Δt always faces many challenges in practice: if this value is too big, a replay attack occurs; if it is too small, a valid participant may be stopped. Therefore, in protocol design, the use of random numbers is usually a more recommended way. Unfortunately, the timestamps method was applied in Amin et al.'s scheme.

Insecure identity update phase: similar to the process in "offline dictionary attack II" of Section 3.2, an adversary can carry out an offline dictionary attack via using DD_i or DD_s as the verification parameter when U_i tries to update his/her identity.

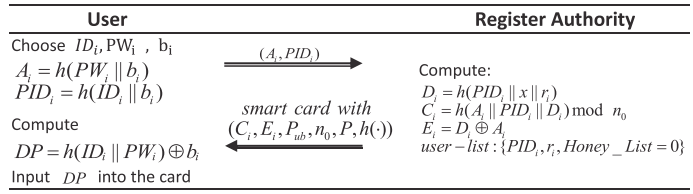

Figure 1: User registration phase.

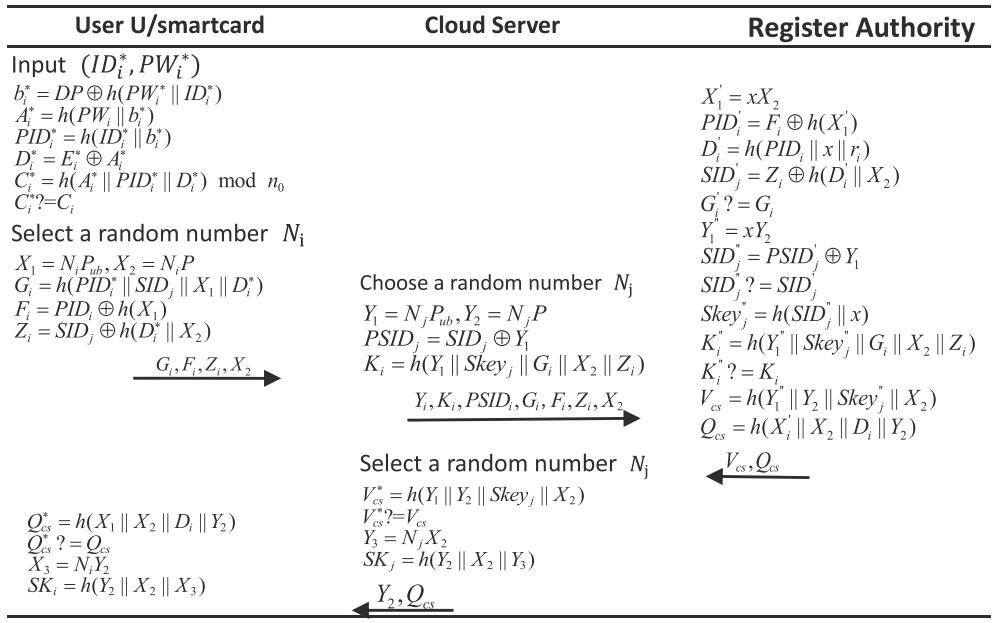

Figure 2: Login and authentication phase.

4. The Proposed Scheme

In this section, we design a secure, simple, and efficient user authentication scheme for cloud computing environment (as shown in Figures 1 and 2) which overcomes all the flaws of Amin et al.'s scheme [3] but provides more attractive attributes, such as updating password and identity and user reregistering with same identity. As a matter of fact, we improve Amin et al.'s scheme from two aspects, efficiency and security. The designing rationales are sketched as follows.

(i) Improvements in Security. According to our discussion in Section 3.2, the public-key algorithm is indispensable to a secure two-factor user authentication scheme [20, 29, 31]. As a matter of fact, this theory has been widely accepted by massive new authentication schemes [26, 30, 35, 36], while the main difficulty lies in deploying the public key algorithm properly. Consequently, we will show the subtleties of deploying a public key algorithm in detail as follows.

Note that, as its high efficiency, the elliptic curve cryptosystem has been used widely in authentication schemes [21, 37–39]. Therefore, our scheme deploys the elliptic curve algorithm to achieve our secure authentication scheme. Under this circumstance, compared with Amin et al.'s scheme [3], our scheme adds a parameter initialization process to set the parameters of elliptic curve cryptosystem.

(1) Apply public-key algorithm to resist against offline dictionary guessing attack II. In Section 3.2, we have shown the pivotal point in addressing such an attack is to add a dynamic nonpublic parameter in G_i. Here we show the ideas about the way of achieving that, to U_i, the shared parameter D_i between RA and U_i is an outcome derived by (ID_i, PW_i). So any parameter based on the transform of (ID_i, PW_i) is futile, as such an parameter is equivalent to D_i. While, with the help of the public-key algorithm, U_i and RA can share a new parameter in every session. More specifically, U_i can choose a random number N_i, compute $X_1 = N_i Pub$, $X_2 = N_i P$ where $Pub = xP$, then let X_1 be the dynamic parameter concealed in G_i. Note that, to U_i and RA, it is easy to compute X_1, yet to $\mathcal{A}$, there is another uncertain parameter in G_i, which stops $\mathcal{A}$ carrying out the offline dictionary attack II.

(2) Apply public-key algorithm to provide user anonymity. With the help of X_1, we conceal the identity related parameter PID_i in F_i as $PID_i \oplus h(X_1)$. Since F_i

is changed with N_i, user anonymity is provided. Note that X_1 is the dynamic parameter we mention in "user anonymity" of Section 3.2.

(3) Apply public-key algorithm to achieve forward secrecy. We set the ECCDH problem in session key to achieve forward secrecy as follows: let SK consist of X_3/Y_3 where $X_3/Y_3 = N_iY_2 = N_iN_jP = N_jX_2$, $Y_2 = N_jP$. As a result, to the one (including RA) who does not know N_i/N_j, computing SK is equivalent to solve the ECCDH problem which cannot be solve in the polynomial time. Therefore, the forward secrecy is achieved.

(4) Following Wang et al.'s way [30] of resisting against offline dictionary attack I, as the detailed explanation on this method can be found in Section IV of [30], Section 6.2.4 of [24], or Section 5.2 of this paper, we do not repeat here.

(ii) Improvements in Efficiency. Note that Amin et al.'s scheme [3] only involves some hash operation, while our scheme deploys a public-key algorithm, so the performance of our scheme is certainly not as efficient as Amin et al.'s scheme. However, except the increased cost of the public-key algorithm, we try to optimize other aspects of the performance in Amin et al.'s scheme through reducing unnecessary parameters or calculations as follows:

(1) Reduce the number of random numbers selected by U_i to one during the user registration process. In Amin et al.' scheme, there are two random numbers in the smart card. While they actually can derive from each other, in addition, the "ability" of computing them is the same, which means they are "equivalent". As a result, using one random number is enough, which saves the storage space and computing resources.

(2) Reduce the number of secret keys in RA. As we see in Section 3.2, the secret key y is of little effect in improving the security. Furthermore, it makes the register phase of CS_j and the authentication of CS_j more complex. Therefore, we only set one system secret parameter and simplify the register phase of CS_j. Such changes also bring other improvements on computing performance.

4.1. Registration. RA selects two large primes $\{p, q\}$ and a medium integer n_0 ($2^4 \le n_0 \le 2^8$). Let $\mathbb{F}_p$ be a finite field, $E/\mathbb{F}_p$ be an elliptic curve over $\mathbb{F}_p$, and $\mathbb{G}$ be a q-order subgroup of $E/\mathbb{F}_p$, then RA chooses a point P in $\mathbb{G}$ and a long-term secret key $x \in Z_p^*$ and computes its public key Pub as xP. In our cloud computing environment, the cloud server and the user registration phases are conducted as follows.

For the Cloud Server CS_j

Step 1. $CS_j \Longrightarrow RA$: $\{SID_j\}$.

Step 2. $RA \Longrightarrow CS_j$: $Skey_j = h(SID_j \parallel x)$.

Step 3. CS_j stores $Skey_j$ as a secret key.

For the UserU_i

Step 1. $U_i \Longrightarrow RA$: $\{A_i, PID_i\}$. A new user U_i firstly selects the password PW_i, identity ID_i, and a random number b_i as his/her personal information and then computes the registration parameters as follows: $A_i = h(PW_i \parallel b_i)$, $PID_i = h(ID_i \parallel b_i)$, and it initiates the registration phase via submitting the request $\{A_i, PID_i\}$ to RA.

Step 2. $RA \Longrightarrow U_i$: a smart card with $\{C_i, E_i, Pub, n_0, h(\cdot)\}$. To guarantee the uniqueness of user identity, RA will firstly check whether the PID_i has been used via traverse $user-list$. If it is available, RA chooses a unique random number r_i for U_i and computes $D_i = h(PID_i \parallel x \parallel r_i)$, $C_i = h(A_i \parallel PID_i \parallel D_i) \bmod n_0$, $E_i = D_i \oplus A_i$, then inputs U_i related parameters $\{PID_i, r_i, Honey\_List = 0\}$ into $User-list$. Note that $Honey\_List$ will record the number of login failures. Finally RA accepts U_i's registration through issuing him/she a smart card with $\{C_i, E_i, Pub, n_0, h(\cdot)\}$.

Step 3. U_i inputs DP into the card where $DP = h(ID_i \parallel PW_i) \oplus b_i$.

4.2. Login phase. As shown in Figure 1, if the user wants to access a cloud server, U_i will submit the information to prove his/her legitimacy in login phase. If the smart card has verified U_i's legality, it initiates the access request to RA for U_i. In short, our login phase involves two aspects: (1) verifying the validity of U_i; (2) initiating the access request.

Step 1. $U_i \longrightarrow CS_j$: login request $\{G_i, F_i, Z_i, X_2\}$. U_i enters $\{ID_i^*, PW_i^*\}$, then the smart card computes $b_i^* = DP \oplus h(ID_i^* \parallel PW_i^*)$, $A_i^* = h(PW_i^* \parallel b_i^*)$, $PID_i^* = h(ID_i^* \parallel b_i^*)$, $D_i^* = E_i \oplus A_i^*$, $C_i^* = h(A_i^* \parallel PID_i^* \parallel D_i^*) \bmod n_0$. The card compares C_i^* with the stored C_i to verify the valid U_i. If $C_i^* \neq C_i$, exit the session.

Step 2. Otherwise, the card accepts U_i's legitimacy and initiates an access request for U_i: select a random number $N_i \in [1, q-1)$, computes $X_1 = N_iPub$, $X_2 = N_iP$, $G_i = (PID_i^* \parallel SID_j \parallel X_1 \parallel D_i^*)$, $F_i = PID_i^* \oplus h(X_1 \parallel X_2)$, $Z_i = SID_j \oplus h(D_i^* \parallel X_2)$, finally transmits $\{G_i, F_i, Z_i, X_2\}$ to CS_j.

4.3. Authentication Phase. Once getting the access request, the CS_j will do some calculation to embed its unique parameters and transmit the request to RA for it is unable to check the authenticity of the request. Then RA will check the validity of the user and the cloud server, respectively, and help them to negotiate the session key. The whole authentication steps are as follows.

Step 1. $CS_j \longrightarrow RA$: $\{Y_2, K_i, PSID_j, G_i, F_i, Z_i, X_2\}$. CS_j selects a random number N_j, computes $Y_1 = N_jPub$, $Y_2 = N_jP$, $PID_j = SID_j \oplus Y_1$, $K_i = h(Y_1 \parallel Skey_j \parallel G_i \parallel X_2 \parallel Z_i)$, then it sends $\{Y_2, K_i, PSID_j, G_i, F_i, Z_i, X_2\}$ to RA.

Step 2. $RA \longrightarrow CS_j$: $\{V_{cs}, Q_{cs}\}$. *RA* will firstly verify U_i via computing the predefined shared parameter D_i and potential shared secret parameters X_1: $X_1' = xX_2$, $PID_i' = F_i \oplus h(X_1')$, $D_i' = h(PID_i' \parallel x \parallel r_i)$ where r_i is retrieved from the $User - list$ through PID_i', $SID_j' = Z_i \oplus h(D_i' \parallel X_2)$. If U_i's $Hoeny - List$ exceeds a predetermined secure value or the received $G_i \neq (PID_i' \parallel SID_j' \parallel X_1' \parallel D_i')$, *RA* views U_i as an adversary and let $Hoeny - List = Hoeny - List + 1$ (if it exceeds the preset value, *RA* will suspend the card till U_i reregisters), then rejects the request.

Otherwise, *RA* will continue to verify the authenticity of the cloud server as follows: calculate $Y_1'' = xY_2$, $SID_j'' = PSID_j \oplus Y_1$. If $SID_j'' \neq SID_j$, exit (*RA* thinks CS_j is not the server which U_i actually desires to access); otherwise, continue to compute $Skey_j'' = h(SID_j'' \parallel x)$, $K_i'' = h(Y_1'' \parallel Skey_j'' \parallel G_i \parallel X_2 \parallel Z_i)$. Finally, *RA* tests the authenticity of CS_j by checking whether $K_i'' \stackrel{?}{=} K_i$. If they are not equal, CS_j does not pass *RA*'s authentication, the session will be terminated.

Otherwise, *RA* authenticates CS_j and then help them to establish the session key as follows: computes $V_{cs} = h(Y_1'' \parallel Y_2 \parallel Skey_j'' \parallel X_2)$, $Q_{cs} = h(X_1' \parallel X_2 \parallel D_i | Y_2)$, then responds $\{V_{cs}, Q_{cs}\}$ to CS_j.

Note that V_{cs} and Q_{cs} are used to CS_j and U_i, respectively. They are used to verify the authenticity of *RA* and then convince CS_j/U_i that X_2/Y_2 is truly generated by U_i/CS_j.

Step 3. $CS_j \longrightarrow U_i$: $\{Y_2, Q_{cs}\}$. CS_j computes $V_{cs}^* = h(Y_1 \parallel Y_2 \parallel Skey_j \parallel X_2)$, then check whether $V_{cs}^* \stackrel{?}{=} V_{cs}$. If the equation is not satisfied, CS_j rejects U_i's request. Otherwise, CS_j believes that U_i is a legitimate user who generates X_2, then CS_j computes the session key as $SK_j = h(Y_2 \parallel X_2 \parallel Y_3)$ where $Y_3 = N_j X_2$. Finally, CS_j transmits $\{Y_2, Q_{cs}\}$ to U_i.

Step 4. On receiving this message, U_i will firstly check the valid of *RA* via comparing $h(X_1 \parallel X_2 \parallel D_i^* \parallel Y_2)$ with the received Q_{cs}. If they are equal, U_i trusts *RA* and CS_j, then he/she computes their shared session key as $X_3 = N_i \cdot Y_2 \ (= N_i \cdot (N_j \cdot P) = Y_3)$, $SK_i = h(Y_2 \parallel X_2 \parallel X_3)$. Till now, the authentication phase finishes successfully.

4.4. Password Change Phase. Considering the security and user friendliness, our password change phase is conducted locally, which guarantees the efficiency. In other words, the user can change his/her password freely even when he/she does not connect the Internet. All in all, our password change phase is performed as follows.

Step 1. U_i enters ID_i, PW_i, and new password PW_i^{new}.

Step 2. The smart card verified U_i's authenticity as step 1 of Section 4.2. If U_i is authenticated, the card will carry out the password change process as step 3; otherwise, the card will reject this request.

Step 3. The smart card computes some new parameters for new password as follows: $A_i^{new} = h(PW_i^{new} \parallel b_i^*)$, $E_i^{new} = E_i \oplus A_i^* \oplus A_i^{new}$, $DP^{new} = h(ID_i \parallel PW_i^{new}) \oplus b_i^*$, $C_i^{new} = h(A_i^{new} \parallel PID_i^* \parallel D_i^*) \bmod n_0$. Note that $b_i^*, A_i^*, D_i^*, PID_i^*$ is acquired in step 2. Finally, replace $\{C_i, E_i, DP\}$ with $\{C_i^{new}, E_i^{new}, DP^{new}\}$.

4.5. Identity Update Phase. Considering the following occasions, a user sets the phone number as his/her identity, then when the phone number is changed, the user may also want to update the identity. Therefore, similar to password change, the user also needs to change the identity in practice, although it happens less often. Thus we provide the identity update phase as follows.

Step 1. U_i enters ID_i, PW_i, and new password ID_i^{new}.

Step 2. The smart card verified U_i's authenticity as step 1 of Section 4.2. If U_i is not authenticated, the card will reject the request; otherwise the identity update process proceeds.

Step 3. As the PID_i stored in *RA* is related to ID_i, the identity update phase involves the interaction with *RA*. On this occasion, the smart card submits $\{X_2, G_i, F_i, NPID_i\}$ to *RA* for requesting update identity, where N_i is a random number, $X_1 = N_i Pub$, $X_2 = N_i P$, $G_i = h(PID_i^* \parallel X_1 \parallel D_i^* \parallel PID_i^{new})$, $F_i = PID_i^* \oplus h(X_1 \parallel X_2)$, $PID_i^{new} = h(ID_i^{new} \parallel b_i^*)$, $NPID_i = PID_i^{new} \oplus h(X_1 \parallel X_2)$.

Step 4. *RA* shall verify the valid of U_i as follows: compute $X_1' = xX_2$, $PID_i' = F_i \oplus h(X_1')$, $D_i' = h(PID_i' \parallel x \parallel r_i)$ where r_i is from $User - list$, $PID_i^{new'} = NPID_i \oplus h(X_1 \parallel X_2)$. If the $Hoeny - List$ exceeds a predetermined value or the received $G_i \neq (PID_i' \parallel X_1' \parallel D_i^* \parallel PID_i^{new'})$, *RA* rejects the request and sets $Hoeny - List = Hoeny - List + 1$. Once it exceeds the preset value, suspend the card.

Otherwise, *RA* updates PID_i with $PID_i^{new'}$ in $User - list$ and sends $\{M_{cs}, ND_i\}$ to U_i where $M_{cs} = h(PID_i^{new'} \parallel PID_i' \parallel D_i' \parallel X_1')$, $D_i^{new} = h(PID_i^{new'} \parallel x)$, $ND_i = D_i^{new} \oplus h(X_1')$.

Step 5. After receiving M_{cs} from *RA*, the smart card authenticates *RA* via testing $M_{cs} \stackrel{?}{=} h(PID_i^{new} \parallel PID_i^* \parallel D_i^* \parallel X_1)$. If the equation does not hold, exit the session; otherwise, finish the identity update process: $DP^{new} = h(ID_i^{new} \parallel PW_i) \oplus b_i^*$, $D_i^{new'} = ND_i \oplus h(X_1)$, $E_i^{new} = D_i^{new'} \oplus A_i^*$, $C_i^{new} = h(A_i^* \parallel PID_i^{new} \parallel D_i^{new'}) \bmod n_0$. Finally, replace $\{C_i, E_i, DP\}$ with $\{C_i^{new}, E_i^{new}, DP^{new}\}$.

4.6. Re-Register Phase. If U_i's smart card is suspended, then he/she shall reregister to *RA*:

Step 1. $U_i \Longrightarrow RA$: $\{A_i, PID_i, re - register\}$.

Step 2. *RA* firstly finds PID_i in $User - list$ and checks whether U_i's card is suspended. If so, *RA* accepts the request and performs the register phase as Section 4.1.

5. Security Analysis

In this section, we analyze the security of our scheme via two popular methods. The results demonstrate that our protocol is secure and effective for the cloud computing environment.

5.1. Formal Analysis Based on BAN Logic. In this section, we apply the BAN logic [40] which is a widely accepted way to analyze the design logic and security of the authentication scheme. Its particular notions to depict protocols are shown in Table 2.

In BAN logic, the goals of our authentication scheme are defined as follows:

(i) Goal 1: $U_i \mid\equiv CS_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} CS_j)$.

(ii) Goal 2: $U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} CS_j)$.

(iii) Goal 3: $CS_j \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} CS_j)$.

(iv) Goal 4: $CS_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} CS_j)$.

According to the proof steps in BAN logic, we redescribe our scheme into an idealized form:

(i) $Message_1$: $U_i \longrightarrow CS_j$: $\langle X_2, G_i, F_i, Z_i, U_i \overset{X_1}{\longleftrightarrow} RA\rangle_{U_i \overset{D_i}{\longleftrightarrow} RA}$.

(ii) $Message_2$: $CS_j \longrightarrow RA$: $\langle Message_1, Y_2, K_i, PSID_j\rangle_{CS_j \overset{Skey_j}{\longleftrightarrow} RA}$.

(iii) $Message_3$: $RA \longrightarrow CS_j$: $\langle X_2, V_{cs}, \langle Q_{cs}\rangle D_i\rangle_{CS_j \overset{Skey_j}{\longleftrightarrow} RA}$.

(iv) $Message_4$: $CS_j \longrightarrow U_i$: $\langle Y_2, \langle Q_{cs}\rangle D_i\rangle_{U_i \overset{D_i}{\longleftrightarrow} RA}$.

Then, some assumptions are defined as follows:

(i) H_1: $U_i \mid\equiv \sharp(X_2)$.

(ii) H_2: $CS_j \mid\equiv \sharp(Y_2)$.

(iii) H_3: $RA \mid\equiv \sharp(X_2)$.

(iv) H_4: $RA \mid\equiv \sharp(Y_2)$.

(v) H_5: $RA \mid\equiv CS_j \overset{Skey_j}{\longleftrightarrow} RA$.

(vi) H_6: $CS_j \mid\equiv CS_j \overset{Skey_j}{\longleftrightarrow} RA$.

(vii) H_7: $RA \mid\equiv U_i \overset{D_i}{\longleftrightarrow} RA$.

(viii) H_8: $U_i \mid\equiv U_i \overset{D_i}{\longleftrightarrow} RA$.

(ix) H_9: $U_i \mid\equiv CS_j \mid\Longrightarrow U_i \overset{SK}{\longleftrightarrow} CS_j$.

(x) H_{10}: $CS_j \mid\equiv U_i \mid\Longrightarrow U_i \overset{SK}{\longleftrightarrow} CS_j$.

Based on the definition above, we perform the BAN logic proof as follows:

From $Message_2$ ($Message_2$ includes $Message_1$), it is easy to get S_1: $RA \triangleleft \langle Message_1, Y_2, K_i, PSID_j\rangle_{D_i}$.

Then according to H_7, S_1, $RULE(1)$, we get S_2: $RA \mid\equiv U_i \mid\sim \langle X_2, G_i, F_i, Z_i, U_i \overset{X_1}{\longleftrightarrow} RA\rangle$.

According to H_3 and $RULE(4)$, we get S_3: $RA \mid\equiv \sharp\langle X_2, G_i, F_i, Z_i, U_i \overset{X_1}{\longleftrightarrow} RA\rangle$.

And according to S_2, S_3 and $RULE(2)$, we get S_4: $RA \mid\equiv U_i \mid\equiv \langle X_2, G_i, F_i, Z_i, U_i \overset{X_1}{\longleftrightarrow} RA\rangle$.

From the Message, we also get S_5: $CS_j \triangleleft \langle Y_2, K_i, PSID_j\rangle_{Skey_j}$.

Then according to H_7, S_1, $RULE(1)$, we get S_6: $CS_j \mid\equiv RA \mid\sim \langle Y_2, K_i, PSID_j\rangle$.

According to H_3 and $RULE(4)$, we get S_7: $CS_j \mid\equiv \sharp\langle Y_2, K_i, PSID_j\rangle$.

And according to S_2, S_3 and $RULE(2)$, we get S_8: $CS_j \mid\equiv RA \mid\equiv \langle Y_2, K_i, PSID_j\rangle$.

From $Message_3$, it is easy to get S_9: $RA \triangleleft \langle X_2, V_{cs}, \langle Q_{cs}\rangle D_i\rangle_{Skey_j}$.

Then according to H_7, S_1, $RULE(1)$, we get S_{10}: $RA \mid\equiv CS_j \mid\sim \langle X_2, V_{cs}, \langle Q_{cs}\rangle D_i\rangle$.

According to H_3 and $RULE(4)$, we get S_{11}: $RA \mid\equiv \sharp\langle X_2, V_{cs}, \langle Q_{cs}\rangle D_i\rangle$.

And according to S_2, S_3 and $RULE(2)$, we get S_{12}: $RA \mid\equiv CS_j \mid\equiv \langle X_2, V_{cs}, \langle Q_{cs}\rangle D_i\rangle$.

From $Message_4$, it is easy to get S_{13}: $U_i \triangleleft \langle Y_2, \langle Q_{cs}\rangle D_i\rangle_{D_i}$.

Then according to H_7, S_1, $RULE(1)$, we get S_{14}: $U_i \mid\equiv RA \mid\sim \langle Y_2, \langle Q_{cs}\rangle D_i\rangle$.

According to H_3 and $RULE(4)$, we get S_{15}: $U_i \mid\equiv \sharp\langle Y_2, \langle Q_{cs}\rangle D_i\rangle$.

And according to S_2, S_3 and $RULE(2)$, we get S_{16}: $U_i \mid\equiv RA \mid\equiv \langle Y_2, \langle Q_{cs}\rangle D_i\rangle$.

As $SK = h(Y_2 \parallel X_2 \parallel N_i \cdot Y_2)$, and combining S_{12}, S_{16}, we get: S_{17}: $U_i \mid\equiv CS_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} CS_j$ (**Goal 1**).

Similarly, as $SK = h(Y_2 \parallel X_2 \parallel N_j \cdot X_2)$, with S_4, S_8, we get S_{18}: $CS_j \mid\equiv U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} CS_j$ (**Goal 3**).

Finally, according to H_2, S_{17}, and $RULE(3)$, we get S_{19}: $U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} CS_j)$ (**Goal 2**).

And according to H_{10}, S_{18}, and $RULE(3)$, we get S_{20}: $CS_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} CS_j)$ (**Goal 4**).

In conclusion, our scheme achieves Goals 1~4, which promises (1)U_i and CS_j have got authenticated mutually and (2) they negotiate the same session key SK.

5.2. Informal Analysis. Looking at the history of protocol designing, due to its simplicity and effectiveness, the heuristic method "still plays an important role" in cryptanalysis of protocols [20], though it does not have a theoretical form and relies on human experience heavily. Therefore, this section gives the security analysis via the heuristic method.

User Anonymity. As we mentioned in Section 3.2, user anonymity contains two aspects, we prove our user anonymity attribute from two points.

(1) $\mathscr{A}$ has no chance to acquire ID_i. In our scheme, the identity is transmitted in a form of F_i where $F_i = PID_i \oplus h(X_1)$, $PID_i = h(ID_i \parallel b_i)$, it is obvious that $\mathscr{A}$ has two challenges in computing ID_i: firstly, computing PID_i from F_i; then guessing ID_i from PID_i. The one only with N_i or x can compute $h(X_1)$ successfully, while $\mathscr{A}$ has no way to get this two parameters. Furthermore, even with PID_i, $\mathscr{A}$ still cannot conduct a guessing attack to compute ID_i for $\mathscr{A}$ does not know b_i.

Table 2: Notations in BAN logic.

$P \mid\equiv X$	P believes X, i.e., the principal P believes the statement X is true.
$P \triangleleft X$	P sees X, i.e., the principal P receives a message that contains X.
$P \mid\Longrightarrow X$	P has jurisdiction over X, i.e., the principal P can generates or computes X.
$P \mid\sim X$	P said X, i.e., the principal P has sent a message containing X.
$\sharp(X)$	X is fresh, i.e., X is sent in a message only at the current run of the protocol, it is usually a timestamp or a random number.
$P \overset{K}{\longleftrightarrow} Q$	K is the shared key for P and Q.
$P \overset{Y}{\rightleftharpoons} Q$	Y is the secret known only to P and Q or some principals trusted by them.
$\langle X \rangle_Y$	X combined with Y, and Y usually is a secret.
$\{X\}_K$	X encrypted with K.
$\dfrac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \triangleleft \{X\}_K}{P \mid\equiv Q \mid\sim X}$ or $\dfrac{P \mid\equiv P \overset{Y}{\rightleftharpoons} Q, P \triangleleft \langle X \rangle_Y}{P \mid\equiv Q \mid\sim X}$	*RULE*(1): the message-meaning rule. This rule will be used in the proving process.
$\dfrac{P \mid\equiv \sharp(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$	*RULE*(2): the nonce-verification rule. This rule will be used in the proving process.
$\dfrac{P \mid\equiv Q \mid\Longrightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$	*RULE*(3): the jurisdiction rule. This rule will be used in the proving process.
$\dfrac{P \mid\equiv \sharp(X)}{P \mid\equiv \sharp(X, Y)}$	*RULE*(4): the freshness-conjuncatenation rule. This rule will be used in the proving process.

(2) $\mathscr{A}$ cannot track the users: as we discussed above, the user's related unique identification is concealed in F_i. It consists of a dynamic parameter X_1 whose value depends on N_i; that is, F_i changes with N_i in every session. So $\mathscr{A}$ can neither links the sessions to a specific user nor tells whether the two sessions are sent by a same user.

Therefore our scheme achieves user anonymity.

Forward Secrecy. Forward secrecy requires that even the lon- term secret key was exposed, the previous session is still secure. In our scheme, the session key $SK = h(Y_2 \parallel X_2 \parallel X_3)$ where $X_3 = N_i Y_2 = N_j X_2 = N_i N_j P$. With the help of x, and the intercepted parameters X_2 and Y_2 from the open channel, if $\mathscr{A}$ wants to compute X_3, then he/she has to solve the ECCDH problem which cannot be finished in the polynomial time. As a matter of fact, once getting x, $\mathscr{A}$ almost has the same capacity with RA. If RA cannot compute SK in the scheme, $\mathscr{A}$ is probably not able to. This again confirms our previous view: for the security consideration, we shall not let RA know SK. All in all, our enhanced scheme provides forward security.

Mutual Authentication. Mutual authentication is the most basic requirement of a user authentication scheme. In our scheme, RA firstly authenticates U_i through G_i which contains their preset shared secret parameter D_i and the dynamic parameter X_1 generated by the public-key algorithm in step 2 of Section 4.2. Then RA authenticates CS_j through K_i with their shared secret value $Skey_j$. In a short, RA authenticates U_i and CS_j after this process.

If both of U_i and G_i are authenticated, RA further computes Q_{cs} and V_{cs}. Then in step 3 of Section 4.2, CS_j checks the validity of RA via V_{cs}. If RA is authentic, then CS_j believes RA's judgment on U_i, which means that CS_j also trusts the legitimacy of U_i. In conclusion, CS_j authenticates RA and U_i via this process.

On receiving Q_{cs}, U_i verifies RA with Q_{cs} in step 4 of Section 4.2. If RA is authenticated, then U_i also believes the validity of CS_j. Therefore, U_i authenticates RA and CS_j.

In conclusion, our scheme achieves mutual authentication.

Privileged Insider Attack. To avoid privileged insider attack, when U_i registers to RA, he/she does not submit the identity or password directly, but a transform of them: $\{A_i, PID_i\}$. Thus the identity and password are protected by b_i. Even the administrator of RA cannot get b_i to conduct an offline dictionary attack to guess the value of ID_i and PW_i. Therefore, our scheme is secure against privileged insider attack.

Offline Dictionary Attack. As we mentioned when analyzing Amin et al.'s scheme [3] in Section 3.2, there are two common offline dictionary attacks. So we consider two kinds of adversary here.

Suppose an adversary $\mathscr{A}$ acquires $\{C_i, E_i, DP, n_0\}$ in the smart card, then $\mathscr{A}$ may conduct an offline dictionary attack as follows.

Step 1. Guess ID_i and PW_i to be ID_i^* and PW_i^*, respectively.

Table 3: Performance comparison among relevant schemes in wireless sensor networks.

	Computation overhead		Communication cost		The evaluation criteria in [24]												
	Login(ms)	Auth.(ms)	Login	Auth.	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
Maitra et al. (2016) [25]	$6T_H \approx 0.04$	$5T_S+14T_H \approx 0.12$	640 bits	1280 bits	√	×	√	×	√	×	×	×	√	√	×	√	√
Kumari et al. (2017) [26]	$2T_M+3T_H \approx 2.3$	$6T_M+15T_H \approx 7.0$	2176 bits	9088 bits	√	√	√	√	×	√	√	×	√	√	√	√	√
Amin et al. (2018) [3]	$6T_H \approx 0.004$	$17T_H \approx 0.012$	640 bits	1920 bits	√	×	√	×	×	√	√	×	√	√	×	√	√
Our scheme	$2T_E+7T_H \approx 1.0$	$4T_E+12T_H \approx 2.0$	1408 bits	4096 bits	√	√	√	√	√	√	√	√	√	√	√	√	√

T_M denotes the time of modular exponentiation operation, T_E denotes scalar multiplication on elliptic curve, T_H denotes hash computation, T_S denotes symmetric encryption/decryption, $T_M \gg T_E \gg T_H > T_S$ ($T_M \approx 1.169ms$, $T_E \approx 0.508ms$, $T_H \approx 0.693\mu s$, $T_S \approx 0.541\mu s$ [20]). Let n_0 be 32-bit long; Let ID_i, PW_i, $h(\cdot)$, output of symmetric encryption, timestamp, random numbers be 128-bit long; let P, x, y be 1024-bit long. √ means the property is satisfied; × means the property is not satisfied. Note that the evaluation criteria in [24] are also applied to cloud computing environment when regarding CS_j as the sensor node, while the considerations on designing authentication scheme of this two environment are quite different due to their different network attributes.

Step 2. Compute $b_i^* = DP \oplus h(ID_i^* \parallel PW_i^*)$ Section 4.1.

Step 3. Compute $A_i^* = h(PW_i^* \parallel b_i^*)$.

Step 4. Compute $PID_i^* = h(ID_i^* \parallel b_i^*)$.

Step 5. Compute $D_i^* = E_i \oplus A_i^*$.

Step 6. Compute $C_i^* = h(A_i^* \parallel PID_i^* \parallel D_i^*) \bmod n_0$.

Step 7. Verify the correctness of PW_i and ID_i by checking if $(C_i^* == C_i)$.

Step 8. Repeat steps 1 ~ 7 until the correct valued of PW_i and ID_i are found.

Now we suppose $\mathscr{A}$ has found such a pair of $\{ID_i^*, PW_i^*\}$ after the above steps. While, due to the properties of equation C_i, even the pair satisfies the equations, it is likely that they are not equal to $\{ID_i, PW_i\}$, since there are $|\mathscr{D}_{pw}| * |\mathscr{D}_{id}|\backslash n_0 \approx 2^{32}$ candidates of $\{ID_i, PW_i\}$ pair when $n_0 = 2^8$ and $|\mathscr{D}_{pw}| = |\mathscr{D}_{id}| = 2^6$ [30]. Consequently, $\mathscr{A}$ needs to verify $\{ID_i^*, PW_i^*\}$ online, but it will be restrained by *Honey_List*. Once the failure numbers of user login exceeds the preset value, the smart card will be suspended. Accordingly, our scheme is secure against such an attack scenario.

Suppose an adversary $\mathscr{A}$ not only extracts the message in smart card, but also eavesdrops $\{G_i, F_i, Z_i, X_2\}$ from the open channel, then $\mathscr{A}$ attempts to guess PW_i and ID_i. In this occasion, $\mathscr{A}$ wants to use G_i as the verification parameter to check the correctness of the guessed value of $\{PW_i, ID_i\}$. As G_i consists of PID_i, SID_j, X_1 and D_i, and according to our attack steps 1~5 above, $\mathscr{A}$ can compute the value of PID_i^* and D_i^*. Then $\mathscr{A}$ computes SID_j^* as $Z_i \oplus h(D_i^* \parallel X_2)$. Now $\mathscr{A}$ only needs to compute X_1. While without x or N_i, computing X_1 is equivalent to solving the ECDL problem which cannot be finished in the polynomial time. As a result, our scheme can prevent such an adversary.

In conclusion, the proposed scheme is secure against dictionary attack.

Verifier-Stolen Attack. In our scheme, *RA* only needs to maintain the *User − list* whose elements ($\{PID_i, r_i, Hoeny\_List\}$) are not security-related. Furthermore, even $\mathscr{A}$ steals *User − list*, he/she will learn nothing useful information to conduct an attack. Thus our scheme is resistant to verifier-stolen attack.

Replay Attack. We prevent the replay attack via the random numbers to prevent replay attack. We take one of the message flow $\{G_i, F_i, Z_i, X_2\}$ as an example to explain: suppose $\mathscr{A}$ eavesdrops $\{G_i, F_i, Z_i, X_2\}$, then replays it to CS_j. While $\mathscr{A}$ does not know N_i, he/she cannot compute the correct session key though the replayed message can pass the verification of *RA*. Consequently, $\mathscr{A}$ gain no benefits from such an attack. Equally, it makes no sense for $\mathscr{A}$ to replay other message flows. Accordingly, our scheme is secure against replay attack.

User Impersonation Attack. According to the definition on user impersonation attack in [24], $\mathscr{A}$ does not acquire the smart card (this condition is included in "offline dictionary attack") here. As we analyzed above, $\mathscr{A}$ with smart card can neither guess ID_i and PW_i nor replay $\{G_i, F_i, Z_i, X_2\}$ to impersonate U_i, let alone the adversary without smart card. So there is only one possible method left: constructing $\{G_a, F_a, Z_a, X_{2a}\}$. To construct this message, $\mathscr{A}$ chooses N_a, computes $\{X_{1a}, X_{2a}\}$, forges PID_a and D_a, calculates $\{G_a, F_a, Z_a, X_{2a}\}$, and finally sends it to *RA*. However, after *RA* gets C_{2a}, and PID_a, *RA* may fail to find such a PID_a in *User − list* or computes a D_a' unequal to D_a, both the two conditions lead to the failure in the authentication of U_i. As a result, *RA* finds that $\mathscr{A}$ is not a legitimate user, the attack fails.

Server Impersonation Attack. According to the above analysis, $\mathscr{A}$ can neither conducts the replay attack to impersonate CS_j/RA nor finds ways to compute $Skey_j/x$, so $\mathscr{A}$ cannot impersonate CS_j/RA.

6. Performance Analysis

Some schemes like [41] which involve only two participants is essentially indistinguishable from the traditional client-server architecture and does not apply to cloud computing environments with multiple servers. Some schemes like [42] which are more concerned with authentication issues between wearable and smart phone belong to entity authentication rather than the user authentication discussed in this article. These schemes are not comparable to ours. Therefore, we only compare those having similar system architectures and application scenarios including [3, 25, 26].

As shown in Table 3, our security performance is obviously superior to other protocols: the proposed scheme

achieves all security requirements, while others have more or less security flaws. More specifically, both the schemes of Maitra et al. [25] and Amin et al. [3] fail to achieve user anonymity and forward secrecy and cannot resist against offline dictionary attack, etc. The best one is Kumari et al.'s scheme [26] which can only provide 11 items of security requirements. In terms of computation or communication performance, as we mentioned in Section 4, the schemes only involving one-way hash operation are inevitable cost less in communication and computation than those deploying the public-key algorithm, but they certainly cannot guarantee the security of authentication. Therefore, among the compared schemes, the schemes of Amin et al. [3] and Maitra et al. [25] certainly cost less communication time and load for they only involve some one-way hash operations. However, sacrificing security to achieve high performance is inadvisable in authentication protocols. As a matter of fact, certain cost is unavoidable for security. Then compared with Kumari et al.'s scheme [26] which is equipped with public-key algorithm, our scheme costs 1ms in login phase and 2ms in authentication phase, while theirs is 2.3ms and 7ms, respectively, our computation overhead is better. Furthermore, our communication cost (1408bits in login phase and 4096 bits in authentication phase) is also lower than theirs (2176 bits and 9088 bits). In conclusion, our scheme with all security attributes is more suitable for cloud computing environment.

7. Conclusion

The rapid development of cloud computing makes people's lives more convenient, but also brings huge security concerns. In order to ensure user's privacy and account security in the cloud environment, a large number of authentication schemes were proposed, but they were subsequently pointed out having one or more flaws. In order to explain the subtleties of designing an authentication protocol in the cloud environment, this paper took Amin et al.'s protocol as a study case to provide ideas for designing secure protocol for cloud environment through elaborating the security weaknesses existing in the protocol and its corresponding solutions. In addition, based on the analysis, we designed a secure authentication protocol, used the BAN logic and heuristic analysis method to prove the security of the protocol. When comparing it with related protocols, we found our scheme has obvious advantages.

Data Availability

There are not any data used in my paper.

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. 2017YFB0801900); the BUPT Excellent Ph.D. Students Foundation under Grant No. CX2017206; and the National Natural Science Foundation of China under Grant No. 61472016.

References

[1] T. Mell and P. Grance, "Draft nist working definition of cloud computing," *National Institute of Standards and Technology*, vol. 6, 2009.

[2] H. Li, F. Li, C. Song, and Y. Yan, "Towards smart card based mutual authentication schemes in cloud computing," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 7, pp. 2719–2735, 2015.

[3] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment," *Future Generation Computer Systems*, 2016.

[4] M. Wazid, A. K. Das, S. Kumari, X. Li, and F. Wu, "Provably secure biometric-based user authentication and key agreement scheme in cloud computing," *Security and Communication Networks*, vol. 9, no. 17, pp. 4103–4119, 2016.

[5] 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.

[6] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Computer Networks*, vol. 101, pp. 42–62, 2016.

[7] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, and M. K. Khan, "A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity," *Security and Communication Networks*, vol. 9, no. 15, pp. 2643–2655, 2016.

[8] D. He, S. Zeadally, N. Kumar, and W. Wu, "Efficient and Anonymous Mobile User Authentication Protocol 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.

[9] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 6, pp. 568–581, 2013.

[10] J. H. Yang, Y. F. Chang, and C. C. Huang, "A user authentication scheme on multi-server environments for cloud computing," in *Proceedings of the ICICS 2013*, pp. 1–4, 2013.

[11] J. H. Yang and P. Y. Lin, "An ID-Based User Authentication Scheme for Cloud Computing," in *Proceedings of the IIH-MSP 2014*, pp. 98–101, 2014.

[12] T.-H. Chen, H.-L. Yeh, and W.-K. Shih, "An advanced ECC dynamic ID-Based remote mutual authentication scheme for Cloud Computing," in *Proceedings of the MUE 2011*, pp. 155–159, 2011.

[13] D. Wang, Y. Mei, C. Ma, and Z. Cui, "Comments on an Advanced Dynamic ID-Based Authentication Scheme for Cloud Computing," in *Proceedings of the WISM 2012*, pp. 246–253, 2012.

[14] K.-P. Xue, P.-L. Hong, and C.-S. Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 195–206, 2014.

[15] M.-C. Chuang and M. C. Chen, "An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," *International Journal of Network Security*, vol. 18, no. 5, pp. 997–1000, 2014.

[16] J.-L. Tsai and N.-W. Lo, "A Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services," *IEEE Systems Journal*, vol. 9, no. 3, pp. 805–815, 2015.

[17] V. Odelu, A. K. Das, S. Kumari, X. Huang, and M. Wazid, "Provably secure authenticated key agreement scheme for distributed mobile cloud computing services," *Future Generation Computer Systems*, vol. 68, pp. 74–88, 2017.

[18] P. Gope and A. K. Das, "Robust Anonymous Mutual Authentication Scheme for n-Times Ubiquitous Mobile Cloud Computing Services," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1764–1772, 2017.

[19] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "On the Design of Provably Secure Lightweight Remote User Authentication Scheme for Mobile Cloud Computing Services," *IEEE Access*, 2017.

[20] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous Two-Factor Authentication in Distributed Systems: Certain Goals Are Beyond Attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2015.

[21] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, "An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 76, pp. 37–48, 2016.

[22] C. Wang, G. Xu, and J. Sun, "A secure and anonymous two-factor authentication protocol in multi-server environment," *Security and Communication Networks*, vol. 2018, 15 pages, 2018.

[23] D. Wang, W. Li, and P. Wang, "Measuring Two-Factor Authentication Schemes for Real-Time Data Access in Industrial Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.

[24] C. Wang, G. Xu, and J. Sun, "An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for wireless sensor networks," *Sensors*, vol. 17, no. 12, article no. 2946, 2017.

[25] T. Maitra, S. H. Islam, R. Amin, D. Giri, M. K. Khan, and N. Kumar, "An enhanced multi-server authentication protocol using password and smart-card: cryptanalysis and design," *Security and Communication Networks*, vol. 9, no. 17, pp. 4615–4638, 2016.

[26] S. Kumari, X. Li, F. Wu, A. K. Das, K.-K. R. Choo, and J. Shen, "Design of a provably secure biometrics-based multi-cloud-server authentication scheme," *Future Generation Computer Systems*, vol. 68, pp. 320–330, 2017.

[27] C. Wang and G. Xu, "Cryptanalysis of three password-based remote user authentication schemes with non-tamper-resistant smart card," *Security and Communication Networks*, vol. 2017, Article ID 1619741, 14 pages, 2017.

[28] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[29] C.-G. Ma, D. Wang, and S.-D. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 2215–2227, 2012.

[30] 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*, 2016.

[31] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions," *Computer Networks*, vol. 73, pp. 41–57, 2014.

[32] D.-J. He, M.-D. Ma, Y. Zhang, C. Chen, and J.-J. Bu, "A strong user authentication scheme with smart cards for wireless communications," *Computer Communications*, vol. 34, no. 3, pp. 367–374, 2011.

[33] Q. Jiang, J. Ma, Z. Ma, and G. Li, "Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems," *Journal of Medical Systems*, 2017.

[34] Y. Lu, L. Li, H. Peng, and Y. Yang, "A Novel Smart Card Based User Authentication and Key Agreement Scheme for Heterogeneous Wireless Sensor Networks," *Wireless Personal Communications*, vol. 96, no. 1, pp. 813–832, 2017.

[35] C. Wang, D. Wang, G. Xu, and Y. Guo, "A lightweight password-based authentication protocol using smart card," *International Journal of Communication Systems*, vol. 30, no. 16, Article ID e3336, 2017.

[36] Q. Jiang, J. Ma, C. Yang, X. Ma, J. Shen, and S. A. Chaudhry, "Efficient end-to-end authentication protocol for wearable health monitoring systems," *Computers and Electrical Engineering*, 2017.

[37] M. K. Khan and D. He, "A new dynamic identity-based authentication protocol for multi-server environment using elliptic curve cryptography," *Security and Communication Networks*, vol. 5, no. 11, pp. 1260–1266, 2012.

[38] A. K. Das, S. Kumari, V. Odelu, X. Li, F. Wu, and X. Huang, "Provably secure user authentication and key agreement scheme for wireless sensor networks," *Security and Communication Networks*, vol. 9, no. 16, pp. 3670–3687, 2016.

[39] D. Wang, H. Cheng, D. He, and P. Wang, "On the Challenges in Designing Identity-Based Privacy-Preserving Authentication Schemes for Mobile Devices," *IEEE Systems Journal*, vol. 12, no. 1, pp. 916–925, 2018.

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

[41] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and N. Kumar, "A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers," *The Journal of Supercomputing*, 2017.

[42] F. Wu, X. Li, L. Xu, S. Kumari, M. Karuppiah, and J. Shen, "A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server," *Computers and Electrical Engineering*, vol. 63, pp. 168–181, 2017.