Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 8131367, 12 pages
https://doi.org/10.1155/2018/8131367

Research Article

An Anonymous Authentication Protocol Based on Cloud for Telemedical Systems

Wenmin Li,[1] Shuo Zhang,[1] Qi Su,[2] Qiaoyan Wen,[1] and Yang Chen[1]

[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *State Key Laboratory of Cryptology, Beijing 100878, China*

Correspondence should be addressed to Shuo Zhang; zs1990@vip.qq.com

Received 10 April 2018; Accepted 2 August 2018; Published 2 September 2018

Academic Editor: Joseph Liu

Telecare medical information systems (TMIS) enable patients to access healthcare delivery services conveniently. With the explosive development occurring in cloud computing and services, storage of personal medical and health information outsourcing to cloud infrastructure has been a potential alternative. However, this has entailed many considerable security and privacy issues. In order to address the security loopholes, we propose a promising solution satisfying the requirements of cloud computing scenarios for telemedical systems. The proposed scheme could provide both data confidentiality and message authenticity while preserving anonymity. Furthermore, the formal security proof demonstrates that the proposed scheme is resistant to various attacks. The performance comparisons show the proposal's workability and it is well suited to adoption in telemedical services.

1. Introduction

With the explosion of cloud computing and services, there has been a growing trend to use the cloud for large-scale data storage and management. It is a new style of computing that offers dynamically scalable network services to external customers delivered over the Internet. Cloud computing provides a powerful underlying architecture for telemedicine, which is an emerging treatment mode for delivering appropriate healthcare services remotely. It facilitates medical practitioners and patients to establish communication over public networks and patients can acquire the medical services via electronic networks conveniently. This will significantly lower the social and economic expenses, while enhancing the medical quality and efficiency.

Cloud computing introduces a new way for medical systems to store and manage medical data, which is complex task. As wearable devices are becoming more and more powerful, patients can obtain their health information timely. They also could upload and access their medical records to the cloud through mobile devices. This can help medical institutions to quickly obtain patients' physical condition in urgent cases for proper medical diagnosis and treatment process. Any delay in the access to medical record at the time of emergency would cause severe errors, which profoundly affects patient's therapeutic process. In the cloud based telecare medical information systems, the cloud database is responsible for storing patients' critical medical data and updates it as the medical treatment availed by the patient. However, the storage of patients' electronic medical records such as personal information, medical records, and physiological parameters in the medical server may result in the exposure of patients' privacy. Cloud computing offers expansively developing prospects of new and better models of healthcare; it also raises some security issues due to new potential ways for data theft. And hence, safeguarding security and patients' privacy in cloud based telecare medical information systems are very significant. Authentication mechanism is a prerequisite to verify the legality of all participants and tackle the illegal access in distributed systems, such as wireless interface systems [1, 2], multiple server architecture based systems [3], smart card based system [4], and mobile radio systems [5, 6]. Furthermore, the anonymous authentication could protect users anonymity and prevent the disclosure of private information [7, 8]. Therefore, a secure authentication protocol is a proper

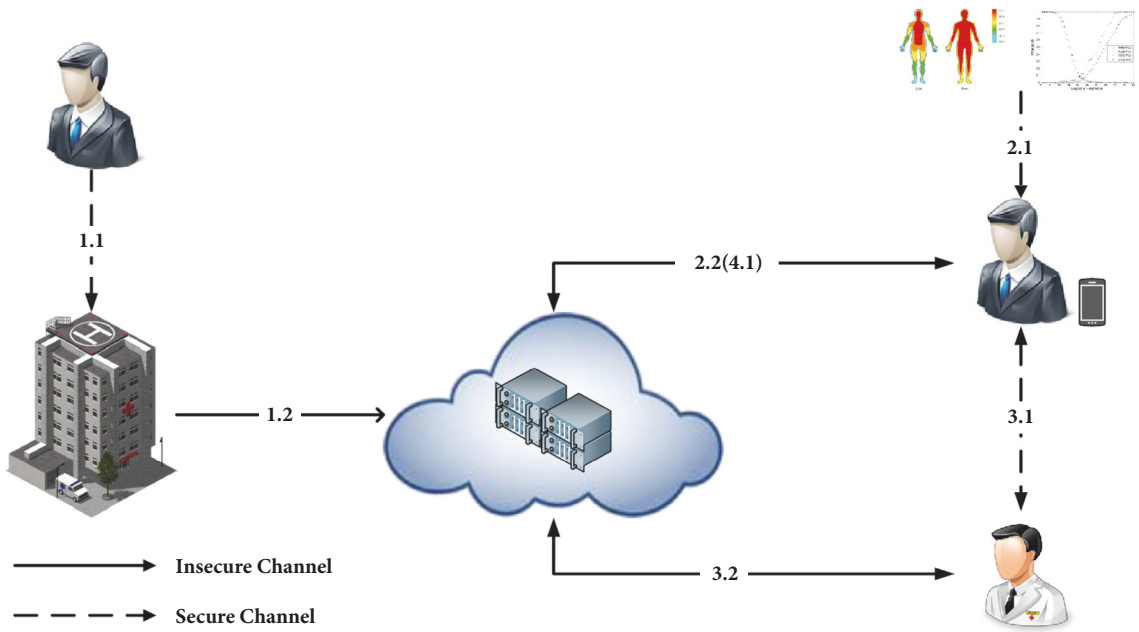

Figure 1: Structure of cloud based TMIS.

solution to provide security and privacy for TMIS [9–11]. Hitherto, authentication protocol for integration telemedical systems in cloud computing environment recently has drawn significant attention from academia [12–20].

In 2012, Padhy et al. [12] introduced a cloud based model for rural healthcare systems. In 2013, Banerjee et al. [13] presented a new architecture for cloud based healthcare application to serve patients in emergency. Nevertheless, their scheme is unable to offer confidentiality of transmitted data. One year later, Chen et al. [14] proposed a medical data exchange protocol in cloud computing environment. In their scheme, patients and doctors could be convenient to access medical resources outsourced in the cloud. Unfortunately, their scheme could not resist impersonation attacks or provide patient anonymity. To fix the defects, a modification was developed in the same year [15]. In 2016, Chiou et al. [16] showed that their scheme still lacks privacy protection and message authentication. Then, the authors proposed a new privacy authentication scheme based on cloud for TMIS which provided a "real" and complete telemedicine system. However, in 2007, Mohit et al. [17], Cheng et al. [20], and Li et al. [18] identified Chiou et al.'s protocol that failed to preserve patients' privacy and forward security and suffers from mobile device stolen attack, respectively. Meanwhile, Mohit et al. [17] and Cheng et al. [20] both presented an improved mechanism for cloud-assisted medical care systems. Recently, Li et al. [19] pointed out that Mohit et al.'s proposal also was susceptible to health report revelation and inspection report forgery attacks. In Cheng et al.'s scheme [20], the inputs of bilinear maps are generators in the corresponding cyclic groups, rather than random numbers of integer field Z_q. This will bring about errors in the authentication process.

In this paper, we design a telemedical information model based on cloud authentication which allows patients to remotely access medical services with privacy. Further, we discuss its security and prove that it can withstand various attacks. Compared with the state of the art, our scheme provides formal security proofs and achieves better efficiency in terms of computation cost. Performance and functionality analysis shows that it is more secure and practical for cloud based telemedicine system.

The remaining of this paper is organized as follows. Section 2 describes our robust cloud based authentication scheme for TMIS, together with formally proving its security in Section 3. Subsequently, we compare the performance with the previous schemes in Section 4. Finally, we draw the conclusions in Section 5.

2. The Proposed Scheme

In this section, we present an anonymous authentication scheme on the basis of cloud for medical environment. There are five participants in our scheme: including patients P, healthcare center H, doctors D, cloud C, and sensors S. Healthcare center is trusted medical center. The cloud servers possess the jurisdiction to store patients' medical data which can be accessed by patients and doctors remotely. Sensors can collect and measure the patient's health information timely. In Figure 1, we depict the structure of the cloud based authentication system for TMIS simply.

Our scheme consists of four phases which are described as follows. In order to initialize this protocol, the key generation center (KGC) chooses a multiplication cyclic group G and a generator $g \in G$ with order p, where p is a large prime number. Then KGC selects random numbers $k_x \in Z_p$ ($x = \{H, P, D, C\}$) and computes $PK_x = g^{k_x} \bmod p$. Finally, KGC issues the public key and secret key pairs (PK_x, k_x) to the participants.

We list the used notations of the proposal as follows.

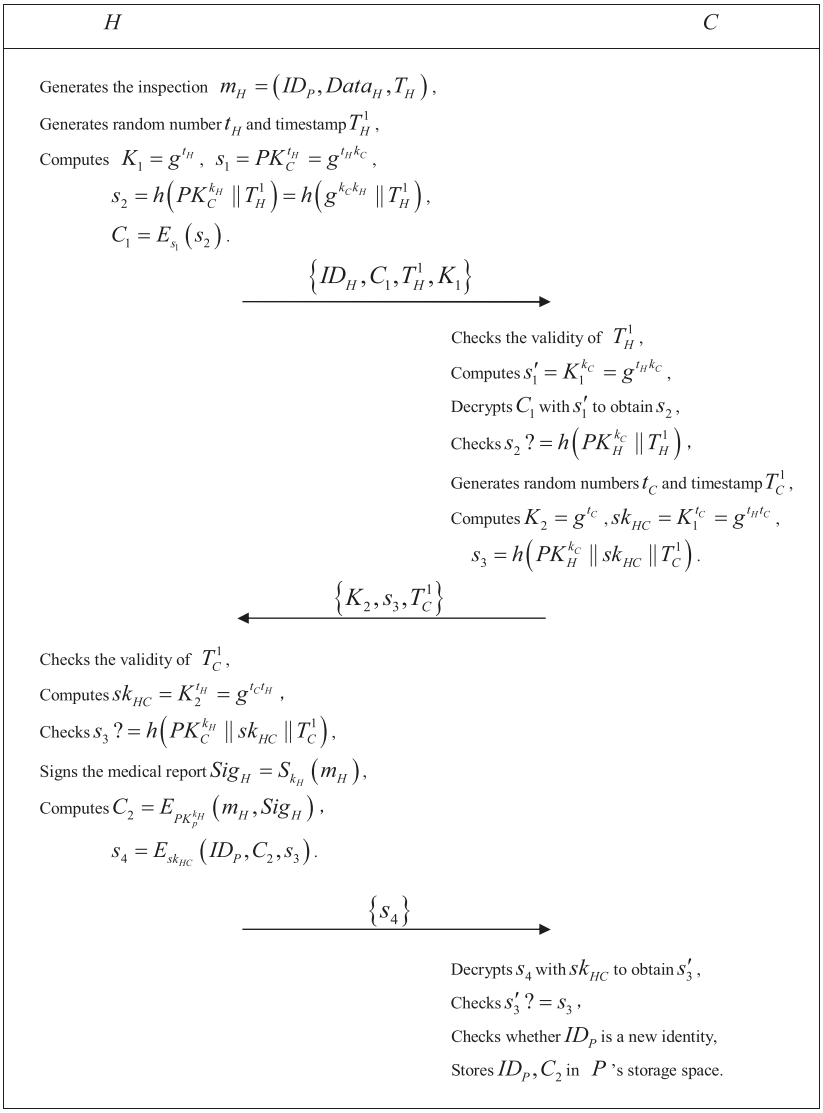

Figure 2: Healthcare center uploading phase.

(i) ID_x: the identity of x

(ii) m_x: the health report of x

(iii) $Data_x$: the health data of x

(iv) sk_{xy}: the session key shared between x and y

(v) PK_x: the public key of x

(vi) k_x: the secret key of x

(vii) Sig_x: signature signed by x

(viii) T_x^i: the ith timestamp generated by x

(ix) G: a multiplicative cyclic group of prime order q

(x) g: the generator of G

(xi) $h(\cdot)$: a one-way hash function

(xii) $\oplus$: exclusive-OR operation

(xiii) $\|$: string concatenation operation

(xiv) $E_{key}(M)$: encrypting M using symmetric encryption with key

(xv) $D_{key}(C)$: decrypting the ciphertext C with key

(xvi) $S_{key}(M)$: signing M with secret key key

(xvii) $V_{key}(Sig_x)$: verifying the signature Sig_x

2.1. Healthcare Center Uploading Phase. As shown in steps 1.1 and 1.2 in Figure 1, patient P makes a health inspection in the healthcare center H and H uploads the generated inspection record to the cloud server C. In Figure 2, we will further describe the authentication process of the phase.

Step 1. After generating the inspection report, H selects a random number t_H and computes $K_1 = g^{t_H}$, $s_1 = PK_C^{t_H} = g^{t_H k_C}$, $s_2 = h(PK_C^{k_H} \| T_H^1) = h(g^{k_C k_H} \| T_H^1)$, $C_1 =$

$E_{s_1}(s_2)$, where T_H^1 is the current timestamp. After that, it sends $\{ID_H, C_1, T_H^1, K_1\}$ to the cloud C.

Step 2. On receiving $\{ID_H, C_1, T_H^1, K_1\}$ from H, C checks the validity of T_H^1. If it is invalid, C aborts the received messages; otherwise, it computes $s_1' = K_1^{k_C} = g^{t_H k_C}$ and decrypts C_1 to recover the values s_2. Subsequently, C calculates $s_2' = h(PK_H^{k_C} \parallel T_H^1)$ and compares it with the decrypted s_2. If the equivalence holds, the legitimacy of H is assured. Then C generates a random number t_C and acquires the timestamp T_C^1 to compute $K_2 = g^{t_C}$, $sk_{HC} = K_1^{t_C} = g^{t_C t_H}$, and $s_3 = h(PK_H^{k_C} \parallel sk_{HC} \parallel T_C^1) = h(g^{k_H k_C} \parallel sk_{HC} \parallel T_C^1)$. Finally, C transmits the authentication message $\{K_2, s_3, T_C^1\}$ to H.

Step 3. Upon receiving the reply message, H checks the validity of T_C^1. If $T_H^2 - T_C^1 \geq \triangle T$, H terminates this phase; otherwise, it computes $sk_{HC} = K_2^{t_H} = g^{t_C t_H}$, $s_3' = h(PK_C^{k_H} \parallel sk_{HC} \parallel T_C^1) = h(g^{k_H k_C} \parallel sk_{HC} \parallel T_C^1)$ and compares s_3' with the received s_3. If they are not equal, the uploading phase is given up by H; else, H signs P's medical report $m_H = (ID_P, Data_H, T_H)$ with its secret key: $Sig_H = S_{k_H}(m_H)$. Note that T_H is the current timestamp when P makes health inspection. After that, H calculates $C_2 = E_{PK_P^{k_H}}(m_H, Sig_H) = E_{g^{k_H k_P}}(m_H, Sig_H)$, $s_4 = E_{sk_{HC}}(ID_P, C_2, s_3)$ and uploads s_4 to the cloud.

Step 4. On receiving s_4, C decrypts it with sk_{HC} to recover ID_P, C_2, s_3' and verifies whether s_3' is equal to s_3 or not. If it is true, the healthcare center is authentic. After that, it verifies whether P is a new user or not. If P is a new user, C stores (ID_P, C_2) in a new storage space; else, it stores C_2 in P's database.

2.2. Patient Uploading Phase. As shown in steps 2.1 and 2.2 in Figure 1, patient P collects health information m_S measured by body sensors S and he could upload the health data to the cloud. In Figure 3, we will depict the detailed process.

Step 1. When P's mobile phone collects the measured information, then it generates the timestamp T_P^1 and a random number a_P to compute $K_3 = g^{a_P}$, $s_5 = PK_C^{a_P} = g^{a_P k_C}$, $s_6 = h(PK_C^{k_P} \parallel T_P^1) = h(g^{k_C k_P} \parallel T_P^1)$, $C_3 = E_{s_5}(ID_P, s_6)$. Subsequently, P transmits $\{C_3, T_P^1, K_3\}$ to C.

Step 2. After receiving the messages, C verifies the freshness of T_P^1 by checking whether $T_C^2 - T_P^1 \leq \triangle T$ or not. If it is valid, C decrypts C_3 to obtain the values ID_P and s_6 with the computed $s_5' = K_3^{k_C} = g^{a_P k_C}$. After that, C calculates $s_6' = h(PK_P^{k_C} \parallel T_P^1)$ and verifies whether the equation $s_6' = s_6$ holds. If it does, P is legitimate user. Then C selects random numbers a_C and computes $K_4 = g^{a_C}$, $sk_{PC} = K_3^{a_C} = g^{a_C a_P}$ and $s_7 = h(PK_P^{k_C} \parallel sk_{PC} \parallel T_C^3) = h(g^{k_P k_C} \parallel sk_{PC} \parallel T_C^3)$, where T_C^3 is the acquired timestamp. Finally, C transmits the $\{K_4, s_7, C_2, T_C^3\}$ to patient P.

Step 3. On receiving response, P checks the validity of T_C^3. If T_C^3 is invalid, P terminates the procedure. If T_C^3 is fresh, P computes $sk_{PC} = K_4^{a_P} = g^{a_C a_P}$, $s_7' = h(PK_C^{k_P} \parallel sk_{PC} \parallel T_C^3) = h(g^{k_P k_C} \parallel sk_{HC} \parallel T_C^3)$ and verifies s_7 is valid by checking whether $s_7' = s_7$ holds. If so, P decrypts C_2 with the computed $PK_H^{k_P}$ to recover $m_H = (ID_P, Data_H, T_H)$ and Sig_H. Subsequently, he/she verifies the validity of H's signature Sig_H. If Sig_H is valid, P chooses a random number sn and computes $C_4 = E_{PK_D^{snk_P}}(m_H, m_S, Sig_H) = E_{g^{snk_D k_P}}(m_H, m_S, Sig_H)$, where $m_S = (ID_P, Data_S, T_S)$ is the collected measured data. Note that T_S is the current timestamp when the body sensors S monitor P's physical condition. Then P calculates $s_8 = E_{sk_{PC}}(ID_P, C_4, s_7)$ and uploads s_8 to the C.

Step 4. On receiving the reply message, C decrypts s_8 with sk_{PC} and obtains ID_P, C_4, s_7'. After that, the cloud server verifies P's validity by checking whether s_7' equals to s_7 or not. If so, C stores C_4 in P's storage space to replace C_2; otherwise, it resumes the procedure.

2.3. Treatment Phase. As shown in steps 3.1 and 3.2 in Figure 1, D is appointed by P and obtains P's identity ID_P and appointment sequence value sn. Subsequently, D can download P's inspection report and measured health information from C, and he/she also can upload the diagnosing records with his/her signature to C. The details of the execution steps are further illustrated in Figure 4.

Step 1. D selects a random number b_D and computes $K_5 = g^{b_D}$, $s_9 = PK_C^{b_D} = g^{b_D k_C}$, $s_{10} = h(PK_C^{k_D} \parallel T_D^1) = h(g^{k_C k_D} \parallel T_D^1)$, $C_5 = E_{s_9}(s_{10})$, where T_D^1 is the acquired current timestamp. Then D transmits $\{ID_D, C_5, T_D^1, K_5\}$ to the cloud C.

Step 2. After receiving messages from D, C checks $T_C^4 - T_D^1 ? \leq \triangle T$. If it is invalid, C terminates the phase; otherwise, it computes $s_9' = K_5^{k_C} = g^{b_D k_C}$ and decrypts C_5 to obtain the values s_{10}. Later, C verifies $s_{10}' = h(PK_D^{k_C} \parallel T_D^1)$ with the decrypted s_{10} to confirm the legitimation of D. If they are equal, C generates random numbers b_C and the timestamp T_C^5 and computes $K_6 = g^{b_C}$, $sk_{CD} = K_5^{b_C} = g^{b_C b_D}$, $s_{11} = h(PK_D^{k_C} \parallel sk_{CD} \parallel T_C^5) = h(g^{k_C k_D} \parallel sk_{CD} \parallel T_C^5)$. Finally, C sends $\{K_6, s_{11}, C_4, T_C^5\}$ to D.

Step 3. On receiving $\{K_6, s_{11}, C_4, T_C^5\}$, D checks the freshness of T_C^5. If so, D computes $sk_{CD} = K_6^{b_D} = g^{b_C b_D}$, $s_{11}' = h(PK_C^{k_D} \parallel sk_{CD} \parallel T_C^5) = h(g^{k_C k_D} \parallel sk_{CD} \parallel T_C^5)$ and compares s_{11}' with the received s_{11} to assure C's authenticity. If they are not equal, the phase is terminated by himself/herself; otherwise, D uses the appointment sequence number sn to compute $PK_P^{snk_D}$ and decrypts C_4 with it to recover m_H, m_S, Sig_H. Subsequently, D verifies whether the signature Sig_H is valid or not by checking $m_H ? = V_{PK_H}(Sig_H)$. If it is valid, D diagnoses P's symptom on the basis of (m_H, m_S) and generates the diagnostic records $m_D = (ID_P, Data_D, T_D)$, where T_D is the timestamp when the doctor generates m_D. After that, D uses his/her private

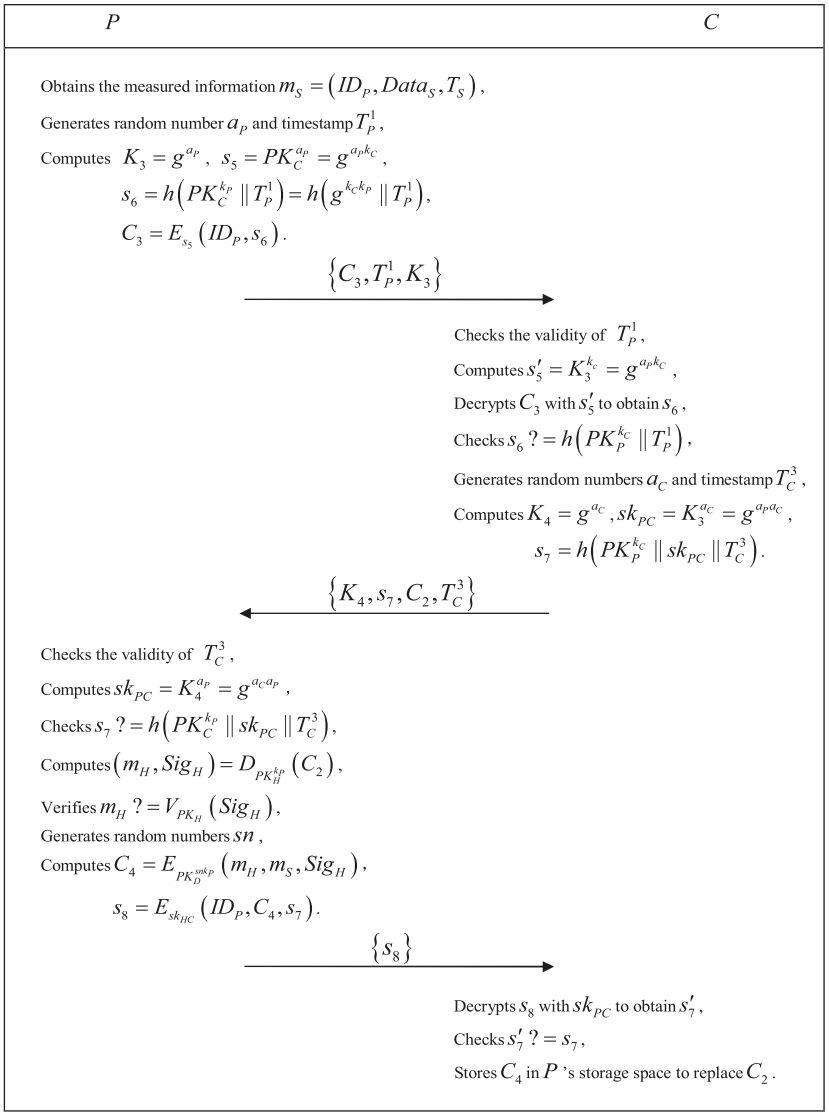

Figure 3: Patient uploading phase.

key k_D to sign m_D: $Sig_D = S_{k_D}(m_D)$. Then, D calculates $C_6 = E_{PK_P^{snk_D}}(m_H, m_S, m_D, Sig_D)$, $s_{12} = E_{sk_{CD}}(ID_P, C_6, s_{11})$ and sends s_{12} to the cloud.

Step 4. Upon receiving s_{12}, C decrypts it with sk_{CD} and obtains ID_P, C_6, s_{11}'. Later, it checks $s_{11}' ? = s_{11}$. If the equation holds, the validity of D is confirmed; otherwise, this phase fails. After that, C replaces C_4 with C_6 and stores it in P's storage space.

2.4. Checking Report Phase. As shown in step 4.1 in Figure 1, patient P can access the cloud to obtain the medical record via the mobile phone. In Figure 5, we depict the detailed process of the phase.

Step 1. P generates the timestamp T_P^2 and a random number v_P to compute $K_7 = g^{v_P}$, $s_{13} = PK_C^{v_P} = g^{v_P k_C}$, $s_{14} = h(PK_C^{k_P} \| T_P^2) = h(g^{k_C k_P} \| T_P^2)$, $C_7 = E_{s_{13}}(ID_P, s_{14})$. Then, P transmits the request $\{C_7, T_P^2, K_7\}$ to the cloud C.

Step 2. C verifies the freshness of T_P^2 after receiving the request from P. If so, C decrypts C_7 with computed $s_{13}' = K_7^{k_C} = g^{v_P k_C}$ and obtains the ID_P, s_{14}'. Subsequently, C calculates $s_{14}' = h(PK_P^{k_C} \| T_P^2)$ and verifies $s_{14}' ? = s_{14}$. If they are equal, C computes $K_8 = g^{v_C}$, $sk_{PC'} = K_7^{v_C} = g^{v_C v_P}$ and $s_{15} = h(PK_P^{k_C} \| sk_{PC'} \| T_C^6) = h(g^{k_P k_C} \| sk_{PC'} \| T_C^6)$, where v_C and T_C^6 are generated random value and the acquired timestamp, respectively. After that, C sends the $\{K_8, s_{15}, C_6, T_C^6\}$ to P.

Step 3. On receiving response $\{K_8, s_{15}, C_6, T_C^6\}$, P checks the validity of T_C^6. If $T_P^3 - T_C^6 \geq \triangle T$, P aborts the session; otherwise, he/she calculates $sk_{PC'} = K_8^{v_P} = g^{v_C v_P}$, $s_{15}' = h(PK_C^{k_P} \|$

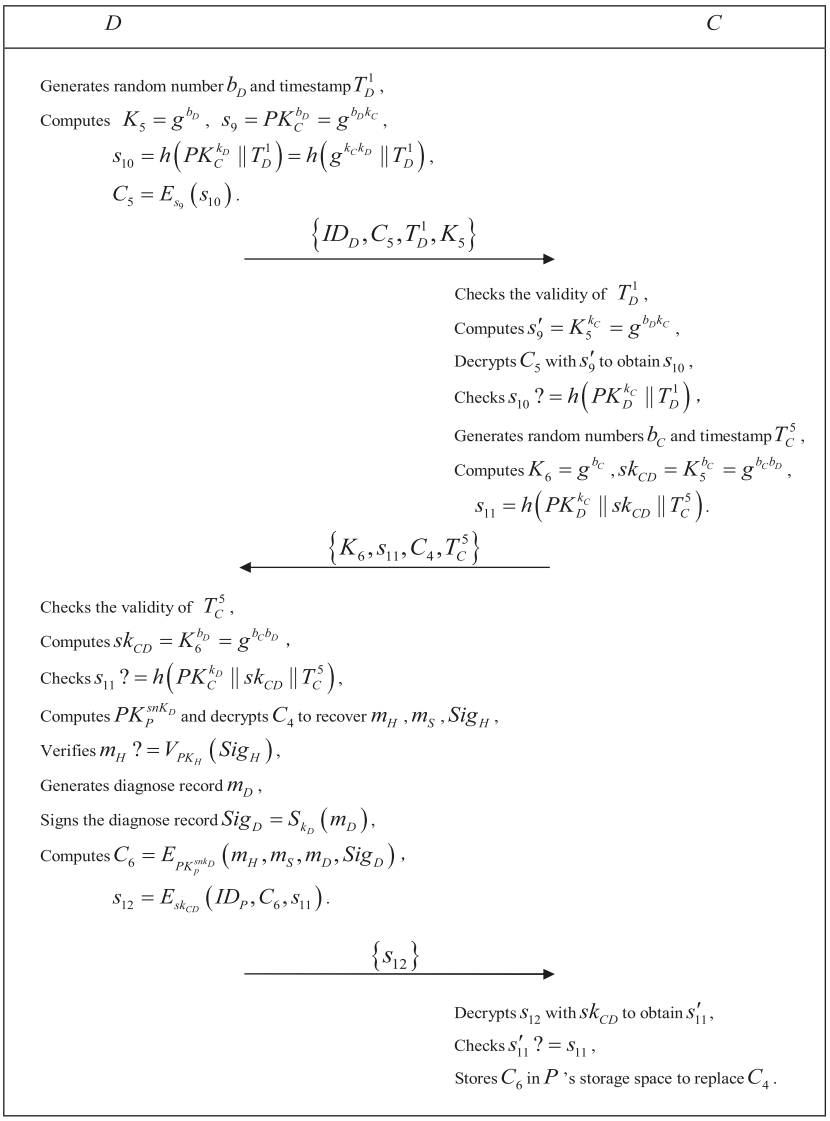

Figure 4: Treatment phase.

$sk_{PC'} \| T_C^6) = h(g^{k_P k_C} \| sk_{PC'} \| T_C^6)$ and verifies whether $s_{15}' = s_{15}$ holds. If yes, P computes $(m_H, m_S, m_D, Sig_D) = D_{PK_D^{snkp}}(C_6)$ and he/she verifies whether $m_D? = V_{PK_D}(Sig_D)$. If so, P takes the diagnosis report of doctor D for the health information m_H and m_S. Finally, P calculates $C_8 = E_{PK_P^{cn}}(m_H, m_S, m_D)$ and $s_{16} = E_{sk_{PC'}}(ID_P, C_8, s_{15})$ and uploads s_{16} to the C, where cn is a pre-generated key to encrypt (m_H, m_S, m_D). Note that cn is the pre-generated key for all the medical staff. If the patient P has an accident, they can decrypt C_8 to obtain the health information and make treatment decisions.

Step 4. Upon receiving the reply s_{16}, C computes $(ID_P, C_8, s_{15}') = D_{sk_{PC'}}(s_{16})$ and verifies whether s_{15}' equals s_{15} or not. If $s_{15}' = s_{15}$, C stores C_8 in P's storage space to replace C_6; otherwise, the phase is failed.

3. Security Proof

In this section, we will prove our scheme to be secure in standard model. We reduce the security of our authentication scheme to cryptography basic elements [21, 22]. At first, in order to achieve this goal, we will introduce the definitions of security, a structured security model, and the basic assumptions. Then we use all of them to prove the result.

Definition 1 (semantic security). For arbitrary security parameter N, if and only if any polynomial time adversary has a negligible advantage against the scheme, we say the scheme has semantic security.

The definitions are inherited and modified from the methodology of Bellare, Pointcheval, and Rogaway [23] and the game-based structure [24] is used to prove this scheme achieving semantic security.

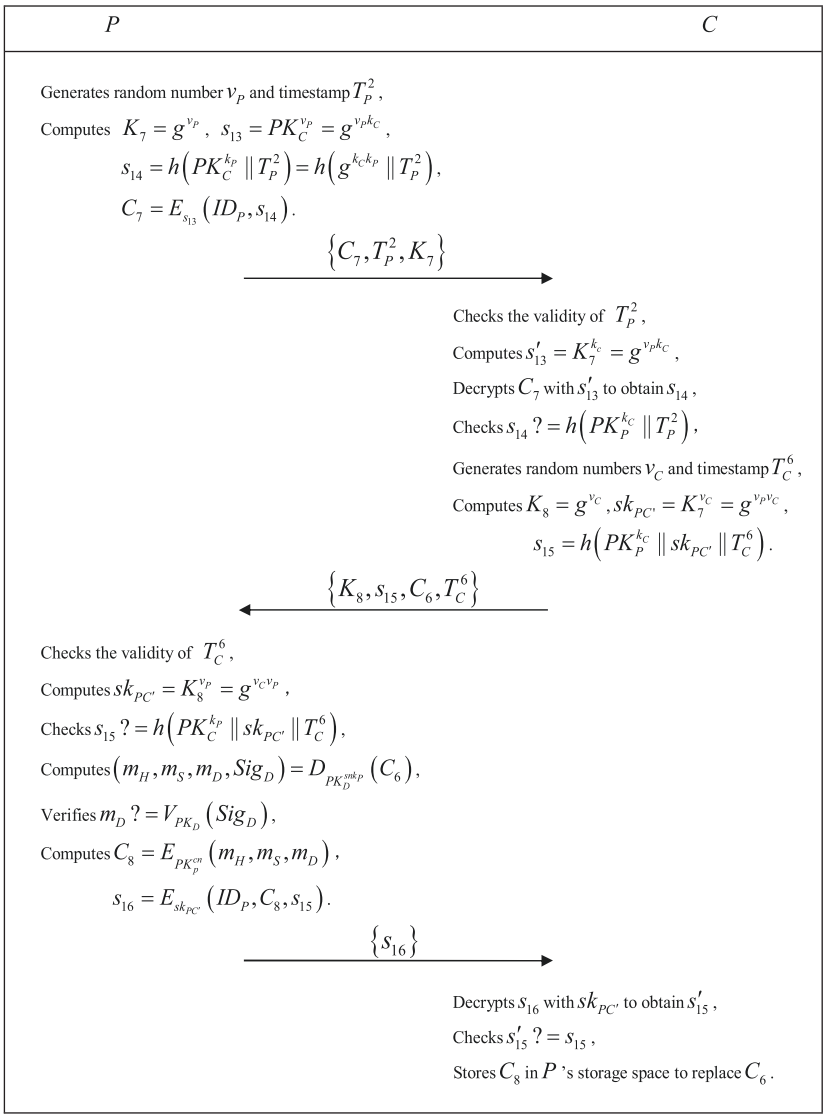

Figure 5: Checking report phase.

Security Model. In the security model, the adversary plays a game with an oracle. The oracle runs the real protocol and answers the queries of adversary to simulate the real interaction of participants. After a range of queries, the adversary gets different capabilities. When the adversary finishes the training and obtains enough messages, oracle should answer the test query once. Finally, we judge if the adversary wins or loses by what the adversary gets. The adversary and the oracle are denoted by $\mathscr{A}$ and $\mathscr{B}$, respectively.

Init: before replying to queries of $\mathscr{A}$, $\mathscr{B}$ generates the system parameters including security parameter N, a multiplication cyclic group G, and a generator $g \in G$ with order p, where p is a large prime number related N. Then $\mathscr{B}$ selects random numbers $k_x \in Z_p$ and computes $PK_x = g^{k_x} \bmod p$ for $x \in \{H, P, D, C\}$. We notice that in a complete system P and D are not unique. Then $\mathscr{B}$ prepares public key and secret key pairs denoted by (PK_{P_i}, k_{P_i}) and (PK_{D_j}, k_{D_j}) for P_i and D_j, where $i \in [1, n]$ and $j \in [1, m]$. $\mathscr{B}$ marks up all P and D with void state. Then it maintains a list of SID recording simulated conversations. $SID^u_{P_i,D_j}$ represents the uth conversation involved P_i and D_j. Noticeably, any SID has a void state before being invoked.

After the init phase, $\mathscr{A}$ is allowed to make queries for simulating the real protocol.

Corrupt(P_i): $\mathscr{B}$ gives k_{P_i} back to $\mathscr{A}$ and marks up the state of P_i with corrupted.

Corrupt(D_j): $\mathscr{B}$ gives k_{D_j} back to $\mathscr{A}$ and marks up the state of D_j with corrupted.

Reveal($SID^u_{P_i,D_j}$): this query simulates abuse of session keys sk.

(1) If the state of $SID^u_{P_i,D_j}$ is void, $\mathscr{B}$ executes a real conversation and gives sk (being used to encrypt m) to $\mathscr{A}$. Then the result of $SID^u_{P_i,D_j}$ is changed into revealed.
(2) Else, if the state of $SID^u_{P_i,D_j}$ is not void, $\mathscr{B}$ answers $\mathscr{A}$ according to the conversation in the list and adds revealed to its state.

Execute($SID^u_{P_i,D_j}$): this query simulates the passive adversary.

(1) When one or both of P_i and D_i are corrupted, $\mathscr{B}$ checks the state of $SID^u_{P_i,D_j}$.
  (a) If void, $\mathscr{B}$ executes a real conversation, gives to $\mathscr{A}$ data transferred over the network, and changes the state of $SID^u_{P_i,D_j}$ into executed.
  (b) Else, if not void, $\mathscr{B}$ answers $\mathscr{A}$ using the conversation in the list and adds executed into its state.
(2) When neither of P_i and D_i is corrupted, $\mathscr{B}$ checks the state of $SID^u_{P_i,D_j}$.
  (a) If void, $\mathscr{B}$ executes a real conversation, gives to $\mathscr{A}$ data transferred over the network, and changes the state of $SID^u_{P_i,D_j}$ into executed.
  (b) Else, if not void, $\mathscr{B}$ answers $\mathscr{A}$ using the conversation in the list and adds executed into its state.

Send(P_i, D_i, Message): this query simulates active attacks.

(1) When one or both of P_i and D_i are corrupted, $\mathscr{B}$ executes a real conversation to answer $\mathscr{A}$.
  (a) If message is verified successfully, $\mathscr{B}$ answers $\mathscr{A}$ as the real protocol and adds this conversation into list with sent state.
  (b) Else, $\mathscr{B}$ outputs that $\mathscr{A}$ is refused.
(2) When neither of P_i and D_i is corrupted, $\mathscr{B}$ answers $\mathscr{A}$ as follows.
  (a) If message is verified successfully, $\mathscr{B}$ answers $\mathscr{A}$ as the real protocol and adds this conversation into list with sent state.
  (b) Else, $\mathscr{B}$ outputs that $\mathscr{A}$ is refused.

Test(m_0, m_1): $\mathscr{B}$ chooses P_i and D_j whose both states are void. Then it randomly chooses a coin b and simulates a real conversation with inputting m_b. $\mathscr{A}$ guesses the result of b.

The complexity assumptions needed for proving security of our scheme are reviewed as follows:

Assumption 2 (DDH). Let G be a cyclic group with order p. $g \in G$ is a generator and $a, b \in Z_p^*$. It is difficult to judge if an element in G equals g^{ab} by just given g^a and g^b.

We denote upper bound of adversaries' advantage against DDH as ϵ_{DDH}. So ϵ_{DDH} should be negligible if the assumption is right.

Assumption 3 (hash). There exists a secure irreversible hash function which achieves strong collision resistance.

We denote the advantage of adversaries against the hash function as ϵ_{hash}. So ϵ_{hash} should be negligible if the assumption is right.

Assumption 4 (signature). There exists a secure digital signature scheme.

We denote advantage of adversaries against this signature scheme as ϵ_{sign}. So ϵ_{sign} should be negligible if the assumption is right.

Assumption 5 (encryption). There is a symmetry encryption that achieves CPA security.

We denote advantage of adversaries against this encryption scheme as ϵ_{enc}. So ϵ_{enc} should be negligible if the assumption is right.

Proof. A PPT adversary $\mathscr{A}$ is attacking the protocol. We use a series of games to bound the advantage of $\mathscr{A}$. The advantage of $\mathscr{A}$ in Game i is defined as

$$Adv_i \overset{def}{=\!=} |2 \cdot Pr\,[\mathscr{A}\ succeeds\ in\ Game\ i] - 1|\,. \quad (1)$$

The games used to bound the advantages of $\mathscr{A}$ are listed in the following. We analyse the advantage difference in nearly games and bound them. In Game 0, it would be the real protocol.

Game 0. $\mathscr{A}$ interacts with the initial security model.

Game 1. In this game, we modify Execute queries. When the states of P_i, D_j, and SID are all void, $\mathscr{B}$ simulates a real protocol but replaces $s_1, sk_{HC}, s_5, sk_{PC}, s_9, sk_{DC}, s_{13}$ and $sk_{PC'}$ with random numbers in G.

Lemma 1. $|Adv_0 - Adv_1| \leq negl$.

Proof. We just replace the sks of traditional DH protocol with random numbers. The advantage difference between two games is caused by DDH problem. And hence, Lemma 1 is proved by DDH assumption right.

Game 2. This game is based on Game 1 and we also modify Execute queries. When the states of P_i, D_j, and SID are all void, $\mathscr{B}$ simulates a real protocol but replaces $s_2, s_3, s_6, s_7, s_{10}, s_{11}, s_{14}, s_{15}$ with uniform random numbers in the range of hash function.

Lemma 2. $|Adv_1 - Adv_2| \leq negl$.

Proof. We just replace the real hash results with random numbers. Without the knowledge of inputs, the probability that $\mathscr{A}$ can distinguish the real hash results and random numbers is less than the advantage of $\mathscr{A}$ that captures the hash. And hence, if the hash function is secure, the probability is negligible.

Table 1: Comparisons of properties.

	Chen et al.'s [15]	Chiou et al.'s [16]	Cheng et al.'s [20]	Ours
Resistance of impersonation attack	Yes	Yes	Yes	Yes
Prevention of replay attack	Yes	Yes	Yes	Yes
Preserving patient privacy	No	Yes	No	Yes
Complete mutual authentication	No	No	No	Yes
Perfect forward secrecy	Yes	Yes	Yes	Yes
Confidentiality	Yes	Yes	No	Yes
Perfect backward secrecy	Yes	Yes	Yes	Yes

Game 3. This game is based on Game 2 and we modify Execute queries. When the states of P_i, D_j, and SID are all void, $\mathscr{B}$ simulates a real protocol but replaces $C_1, C_2, s_4, C_3, C_4, s_8, C_5, C_6, s_{12}, C_7, C_8, s_{16}$ with uniform random numbers in the range of encryption.

Lemma 3. $|Adv_2 - Adv_3| \leq negl$.

Proof. We just replace the symmetric encryption results with random numbers. Without knowing inputs, the probability that $\mathscr{A}$ can distinguish the real encryption results and random numbers is less than the advantage of $\mathscr{A}$ that captures the CPA security symmetric encryption. Thereby, if the symmetric encryption is secure, the probability is negligible.

Game 4. This game is based on Game 3 and we modify Send queries. When the states of P_i and D_j are both void, $\mathscr{B}$ simulates a real protocol but replaces $s_1, sk_{HC}, s_5, sk_{PC}, s_9, sk_{DC}, s_{13}$ and $sk_{PC'}$ with random numbers in G.

Lemma 4. $|Adv_3 - Adv_4| \leq negl$.

Proof. We just replace the sks of traditional DH protocol with random numbers. The advantage difference between two games is caused by DDH problem. Therefore, Lemma 4 is proved by DDH assumption right.

Game 5. This game is based on Game 4 and we modify Send queries. When the states of P_i and D_j are both void, $\mathscr{B}$ simulates a real protocol but replaces $s_2, s_3, s_6, s_7, s_{10}, s_{11}, s_{14}, s_{15}$ with uniform random numbers in the range of hash function.

Lemma 5. $|Adv_4 - Adv_5| \leq negl$.

Proof. We just replace the real hash results with random numbers. Without knowing inputs, the probability that $\mathscr{A}$ can distinguish the real hash results and random numbers is less than the advantage of $\mathscr{A}$ that captures the hash. If the hash function is secure, the probability is negligible.

Game 6. This game is based on Game 5 and we also modify Send queries. When the states of P_i and D_j are both void, $\mathscr{B}$ simulates a real protocol but replaces $C_1, C_2, s_4, C_3 C_4, s_8, C_5, C_6, s_{12}, C_7, C_8, s_{16}$ with uniform random numbers in the range of encryption.

Lemma 6. $|Adv_5 - Adv_6| \leq negl$.

Proof. We just replace the encryption results by random numbers. Without the knowledge of inputs, the probability that $\mathscr{A}$ can distinguish the real encryption results and random numbers is less than the advantage of $\mathscr{A}$ that captures the CPA security symmetric encryption. If the symmetric encryption is secure, the probability is negligible.

In Game 6, we notice that P_i and D_j of test query neither are corrupted. For any SID without being corrupted, the Execute and Send queries are all randomized. So in Game 6, the advantage of $\mathscr{A}$ is zero. So, we can compute the Adv_0 as follows:

$$Adv_0 = \left|Adv_0 - Adv_6\right| \leq \epsilon_{DDH} + \epsilon_{hash} + \epsilon_{enc}, \quad (2)$$

which is a negligible value. □

4. Performance and Functionality Analysis

Herein, we evaluate the performance and functionality of the proposed scheme and compare it with three related schemes for cloud based telemedicine systems, including Chen et al.'s scheme [15], Chiou et al.'s scheme [16], and Cheng et al.'s scheme [20].

The comparisons on the key security properties among these systems are given in Table 1. It is visible that our scheme could achieve all security properties and it is superior to the rest three related schemes. Chen et al.'s scheme [15] fails to provide anonymity and complete mutual authentication, while Chiou et al.'s scheme [16] could not achieve the complete mutual authentication. Furthermore, Cheng et al.'s scheme [20] could not preserve users' privacy, complete mutual authentication, and confidentiality. Note that the proposed scheme offers important security features and it is better suitable for cloud based telemedicine environment.

Meanwhile, we present the comparisons of efficiency in terms of computation loading among these schemes in Table 2. Compared with the other three related schemes, the proposed scheme needs not perform the bilinear pairing and could provide more additional security features. Furthermore, our scheme achieves the provably security in the standard model.

More detailed efficiency comparisons are shown in Figures 6 and 7. We implement the cloud of authentication schemes for cloud based telemedicine systems in Python 3.5.2 using an Intel(R) Core(TM) i5-4590 CPU @ 3.30GHZ with 3300MB RAM and Ubuntu 16.04 system. The simulations of platform for healthcare center, patients, and doctors are

Table 2: Comparisons of computation loading.

		Chen et al.'s [15]	Chiou et al.'s [16]	Cheng et al.'s [20]	Ours
HUP	H	$T_s+T_a+2T_m+2T_p+T_E+3T_h$	$T_s+2T_p+2T_E+4T_h$	$2T_p+2T_E+4T_h+T_s+T_m$	$5T_e + 2T_h + T_s + 3T_E$
	C	$2T_m + 2T_p + T_E + 3T_h$	$T_p + T_E + 3T_h$	$3T_h + T_p + T_E$	$4T_e + 2T_h + 2T_E$
PUP	P	$T_s+T_a+2T_m+2T_p+2T_E+3T_h$	$T_s+3T_p+2T_E+7T_h$	$3T_p+2T_E+6T_h+T_s+T_m$	$T_s + 6T_e + 4T_E + 2T_h$
	C	$2T_m + 2T_p + T_E + 3T_h$	$T_p + 4T_h$	$T_p + 5T_h$	$4T_e + 2T_E + 2T_h$
TP	D	$2T_s+2T_m+2T_p+3T_E+3T_h$	$2T_s+2T_p+2T_E+4T_h$	$2T_s + 2T_E + 2T_p + 5T_h$	$5T_e+4T_E+2T_h+2T_s$
	C	$2T_m+2T_p+T_E+3T_h$	$T_p + 4T_h$	$T_p + 4T_h$	$4T_e+2T_E+2T_h$
CP	P	N/A	$T_s+T_p+2T_E+4T_h$	$T_s+T_p+2T_E+4T_h$	$6T_e+4T_E+2T_h+T_s$
	C	N/A	T_p+4T_h	T_p+4T_h	$4T_e+2T_E+2T_h$

HUP: healthcare center uploading phase.
PUP: patient uploading phase.
TP: treatment phase.
CP: checking report phase.
T_h: time consumption for executing a hash function.
T_E: time consumption for executing the symmetric encryption/decryption operation.
T_e: time consumption for executing a modular exponent operation.
T_p: time consumption for executing a bilinear pairing operation.
T_s: time consumption for executing/verifying a signature.
T_a: time consumption for executing an asymmetric encryption/decryption operation.
T_m: time consumption for executing a multiplication operation.

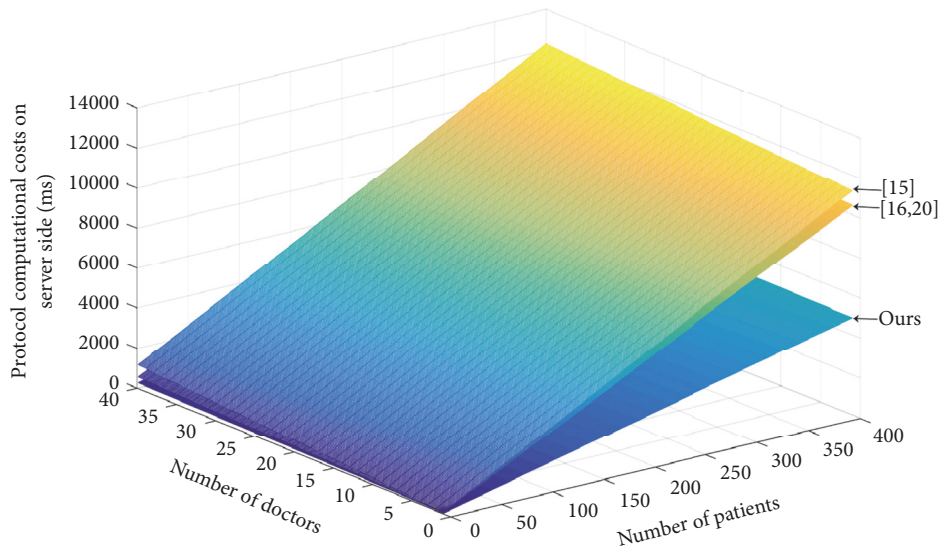

Figure 6: Cost time on cloud computing.

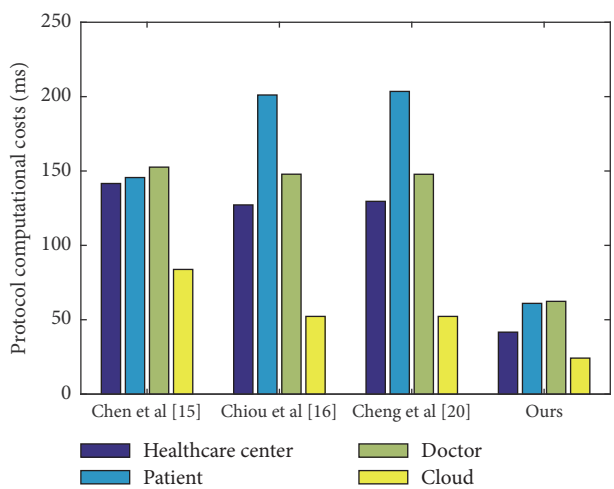

Figure 7: Cost time on healthcare center, patients, doctors, and cloud.

implemented in Python 3.5.2 using an Intel(R) Core(TM) i5-4590 CPU at 1.65GHZ with 1540MB RAM and Ubuntu 16.04 system. The one-way hash function used is SHA-256, and the symmetric encryption/decryption algorithm is advanced encryption standard. We use the ElGamal signature scheme and ElGamal encryption scheme with 1024-bit security parameter for digital signature algorithm and the asymmetric encryption/decryption algorithm, respectively. Moreover, the bilinear paring is simulated in two MNT asymmetric groups, "MNT224".

Figure 6 shows the main cost on the cloud computing of interacting with multiple patients and doctors for authentication simultaneously. It demonstrates that our proposal costs less time for the cloud to authenticate doctors and patients. Figure 7 illustrates the main cost on healthcare center, patients, doctors, and cloud for one round authentication in healthcare center uploading phase, patient uploading phase, treatment phase, and checking report phase, respectively. From Figure 7 we can conclude that our scheme is the most efficient to finish one round mutual authentication.

5. Conclusion

In this article, we proposed an anonymous authentication scheme based on cloud for medical environment, which provided both data confidentiality and message authenticity. Subsequently, we stated that the proposed scheme was provably secure in the standard model. The comparisons with existing competitive protocols also observe that our scheme is suitable for the cloud based telecare medical information systems.

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 work is supported by NSFC (Grant Nos. 61602045, 61502044).

References

[1] W. Shi and P. Gong, "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography," *International Journal of Distributed Sensor Networks*, vol. 9, no. 4, Article ID 730831, pp. 51–59, 2013.

[2] Q. Jiang, J. Ma, G. Li, and L. Yang, "An efficient ticket based authentication protocol with unlinkability for wireless access networks," *Wireless Personal Communications*, vol. 77, no. 2, pp. 1489–1506, 2014.

[3] D. Guo and F. Wen, "Analysis and improvement of a robust smart card based-authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 78, no. 1, pp. 475–490, 2014.

[4] W. Shi and D. He, "A security enhanced mutual authentication scheme based on nonce and smart cards," *Journal of the Chinese Institute of Engineers*, vol. 37, no. 8, pp. 1090–1095, 2014.

[5] D. He, N. Kumar, M. K. Khan, and J.-H. Lee, "Anonymous two-factor authentication for consumer roaming service in global mobility networks," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 4, pp. 811–817, 2013.

[6] 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.

[7] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Information Sciences*, vol. 321, Article ID 11403, pp. 263–277, 2015.

[8] Q. Jiang, F. Wei, S. Fu, J. Ma, G. Li, and A. Alelaiwi, "Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy," *Nonlinear Dynamics*, vol. 83, no. 4, pp. 2085–2101, 2016.

[9] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Systems Journal*, vol. 12, no. 1, pp. 64–73, 2018.

[10] Q. Jiang, M. K. Khan, X. Lu, J. Ma, and D. He, "A privacy preserving three-factor authentication protocol for e-Health clouds," *The Journal of Supercomputing*, vol. 72, no. 10, pp. 3826–3849, 2016.

[11] D. He, N. Kumar, H. Wang, L. Wang, R. K. Choo, and A. Vinel, "A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 633–645, 2016.

[12] R. P. Padhy, M. R. Patra, and S. C. Satapathy, "Design and implementation of a cloud based rural healthcare information system model," *American Journal of Computer Science and Technology*, vol. 2, no. 1, pp. 149–157, 2012.

[13] A. Banerjee, P. Agrawal, and R. Rajkumar, "Design of a cloud based emergency healthcare service model," *International Journal of Applied Engineering Research*, vol. 8, no. 19, pp. 2261–2264, 2013.

[14] C.-L. Chen, T.-T. Yang, and T.-F. Shih, "A secure medical data exchange protocol based on cloud environment," *Journal of Medical Systems*, vol. 38, no. 9, article 112, 2014.

[15] C.-L. Chen, T.-T. Yang, M.-L. Chiang, and T.-F. Shih, "A privacy authentication scheme based on cloud for medical environment," *Journal of Medical Systems*, vol. 38, article 143, 2014.

[16] S. Y. Chiou, Z. Ying, and J. Liu, "Improvement of a privacy authentication scheme based on cloud for medical environment," *Journal of Medical Systems*, vol. 40, no. 4, pp. 1–15, 2016.

[17] P. Mohit, R. Amin, A. Karati, G. P. Biswas, and M. K. Khan, "A stand mutual authentication protocol for cloud computing based health care system," *Journal of Medical Systems*, vol. 41, no. 4, pp. 1–13, 2017.

[18] C. T. Li, D. H. Shih, and C. C. Wang, "On the security of a privacy authentication scheme based on cloud for medical environment," in *International Conference on Information Science and Applications*, vol. 424 of *Lecture Notes in Electrical Engineering*, pp. 241–248, Springer, Singapore, Singapore, 2017.

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

[20] X. F. Cheng, X. L. Zhang, and J. F. Ma, "ICASME: an improved cloud-based authentication scheme for medical environment," *Journal of Medical Systems*, vol. 41, no. 3, pp. 1–14, 2017.

[21] 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.

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

[23] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology—Eurocrypt 2000*, vol. 1807 of *Lecture Notes in Computer Science*, pp. 139–155, Springer, Berlin, Germany, 2000.

[24] A. Groce and J. Katz, "A new framework for efficient password-based authenticated key exchange," in *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS '10)*, pp. 516–525, ACM, Chicago, Ill, USA, October 2010.