Cryptanalysis and Improvement of a Smart Card Based Mutual Authentication Scheme in Cloud Computing

Qi Jiang[1(✉)], Bingyan Li[1], Jianfeng Ma[1], Youliang Tian[2], and Yuanyuan Yang[3]

[1] School of Cyber Engineering, Xidian University, Xi'an, China
jiangqixdu@gmail.com
[2] Guizhou Provincial Key Laboratory of Public Big Data, Guiyang, Guizhou, China
[3] The Third Research Institute of Ministry of Public Security, Shanghai, China

Abstract. Cloud computing enables the users to access and share the data as and when required at anytime from anywhere. Due to its open access, one of the major issues faced by cloud computing is how to prevent the outsourced data from being leaked to unauthorized users. Therefore, mutual authentication between the user and the cloud service provider is a necessity to ensure that sensitive data in the cloud are not available to illegal users. Recently, Li et al. proposed a two-factor authentication protocol based on elliptic curve cryptosystem which enables the cloud users to access their outsourced data. However, we first show that their scheme suffers from the problem of wrong password login. Secondly, their scheme is prone to denial of service attack in the password-changing phase. Thirdly, it fails to provide user revocation when the smart card is lost or stolen. To remedy these flaws, we propose an improved two-factor authentication and key agreement protocol, which not only guards various known attacks, but also provides more desired security properties.

Keywords: Authentication · Key agreement · Password · Smart card · Privacy · Cloud computing

1 Introduction

Cloud computing is a promising computing paradigm where computing and storage resources are provided by third-party service providers with remarkable cost reduction [1–4]. Users are relieved from the cost of buying and maintaining hardware and software platforms. Besides, the users can access and share the data as and when required at anytime from anywhere. However, one of the major issues hindering the adoption of cloud computing is the privacy of outsourced data in cloud may be leaked to unauthorized users, including malicious insiders and outsiders, which renders authentication mechanisms of crucial importance [5–9].

Choudhur et al. [10] proposed a smart card and password based user authentication framework for cloud computing. Hao et al. proposed a time-bound ticket-based mutual authentication scheme for cloud environment using smart card [11]. Although

X. Sun et al. (Eds.): ICCCS 2016, Part I, LNCS 10039, pp. 311–321, 2016.
DOI: 10.1007/978-3-319-48671-0_28

Hao et al. claimed that the authentication scheme was secure; Pippal et al. found it was vulnerable to Denial-of-Service (DOS) attack and the password change phase was insecure [12]. To resist these weaknesses, Pippal et al. proposed an enhancement to Hao et al.'s scheme. Jiang et al. proposed a three-factor authentication scheme for healthcare cloud [13].

However, these authentication protocols [10–13], which are designed for single server environment, are not suitable for multiple-server cloud environment in which a user generally access different types of cloud services from different cloud servers. A user needs to remember the pairs of identity and password corresponding to different servers, the mechanism will bring about a lot of inconvenience to users. Therefore, in the multiple-server cloud environment, it is preferable for a user to use a single pair of identity and password to access different servers. Hwang and Sun proposed a single sign-on scheme for multiple cloud services [14]. Tsai and Lo proposed a privacy aware authentication scheme for distributed mobile cloud computing services [15]. However, Jiang et al. identified that their scheme is vulnerable to the service provider impersonation attack [16]. Recently, Li et al. [17] proposed a two-factor multi-server authentication protocol based on elliptic curve cryptosystem (ECC) which enables the cloud users to access their outsourced data across multiple cloud servers.

However, we find that Li et al.'s protocol is flawed. Specifically, we first show that their scheme suffers from the problem of wrong password login. Secondly, their scheme is prone to DOS attack in the password-changing phase. Thirdly, it fails to provide user revocation when the smart card is lost or stolen. Then, we put forward a robust two-factor authentication scheme. Our new scheme makes up the missing security features necessary for cloud computing while maintaining the desired features of the original scheme. We show that the proposed scheme can withstand various known attacks and provide more security features than Li et al.'s scheme.

The remainder of this paper is organized as follows. The next section briefly reviews Li et al.'s scheme. Section 3 elaborates on the flaws of their scheme. Section 4 presents the improved authentication scheme. In Sect. 5, the security and efficiency of the proposed scheme is analyzed and compared. Finally, the paper is concluded.

2 Review of Li et al.'s Scheme

We review Li et al.'s authentication protocol based on ECC [17], which is composed of three phases, i.e., registration, authentication, and password update. The elliptic curve equation is defined in the form: $E_p(a,b) : y^2 = x^3 + ax + b \,(\mathrm{mod}\, p)$ over a prime finite field F_p, where $a, b \in F_p$, and $4a^3 + 27b^2 \neq 0 (\mathrm{mod}\, p)$ [18]. The notations used in the paper are listed in Table 1.

2.1 Registration Phase

The registration phase involves users and the cloud service provider. When a user A wants to get cloud services, he needs to register in the service provider SP.

Table 1. Notations

Notation	Description
G	A group with order q
P	The generator of G
l	The security length parameter for hash values and random numbers
q	A large prime
A	A user
CS_i	The cloud server
SP	The service provider
ID_A	A's identity
PW_A	A's password
s	SP's private key
sP	SP's public key
K_{CS_i-SP}	The key shared between CS_i and SP
$h(\cdot)$	One-way hash functions: $\{0,1\} \rightarrow \{0,1\}^l$
K_{CS_i-A}, K_{A-CS_i}	The session key established between A and CS_i
$E_k(\cdot)/D_k(\cdot)$	The symmetric encryption/decryption algorithm with a key k
$\|\|$	The concatenation operation;
$\oplus$	The bitwise XOR operation

Step 1. User A first selects PW_A as his/her password. Then, A chooses a random number $r \in Z_q^*$ and computes $h(PW_A||r)$. A sends $\{ID_A, h(PW_A||r)\}$ to the service provider SP through a secure channel.

Step 2. When receiving the message, SP selects a random value $R \in \{0,1\}^{64}$ and computes $C_A = h(s||ID_A||R) \oplus h(PW_A||r)$ for A. SP maintains the value R in database and issues a smart card which contains $\{ID_A, C_A\}$ to A.

Step 3. When receiving the smart card, A stores r into the card. The security parameters in the smart card are $\{ID_A, r, C_A\}$.

2.2 Authentication and Key Exchange Phase

When A wants to get the cloud service, he/she needs to complete a mutual authentication and key exchange with the ith cloud server CS_i.

Step 1. $A \rightarrow CS_i$

A inserts his/her card and inputs his/her password PW_A. Then, he/she selects two random values, $a, r_1 \in Z_q^*$. A computes $K = h(a \cdot sP)$ and $M_A = h(K||r_1) \oplus ID_A$. Next, A reveals $X_A = h(PW_A||r) \oplus C_A$, computes $N_A = h(K||r_1||X_A)$ as the authentication message, and sends $\{aP, r_1, M_A, N_A\}$ to the cloud server CS_i.

Step 2. $CS_i \rightarrow SP$

On receiving $\{aP, r_1, M_A, N_A\}$, CS_i selects a random value $b \in Z_q^*$ and computes $M_{CS_i} = E_{K_{CS_i-SP}}(aP, bP, r_1, M_A, N_A)$. Then, CS_i sends $\{ID_{CS_i}, M_{CS_i}\}$ to the service provider SP.

Step 3. $SP \rightarrow CS_i$

On receiving $\{ID_{CS_i}, M_{CS_i}\}$, SP first decrypts M_{CS_i} and obtains $\{aP, bP, r_1, M_A, N_A\}$. Then, SP computes $K = h(s \cdot aP)$ and $ID_A = h(K||r_1) \oplus M_A$. SP computes $X_A = h(s||ID_A||R)$. After that, SP verifies whether $N_A = h(K||r_1||X_A)$ holds. If it does, SP rejects it. Otherwise, SP selects a random value $s_1 \in Z_q^*$, and computes $Auth_{SP} = h(K||s_1||aP)$ and $M_{SP} = E_{K_{CS_i-SP}}(ID_A, aP, bP, s_1, Auth_{SP})$. SP sends M_{SP} to CS_i.

Step 4. $CS_i \rightarrow A$

On receiving M_{SP}, CS_i first decrypts M_{SP} and obtains $\{ID_A, aP, bP, s_1, Auth_{SP}\}$. Then CS_i verifies whether bP is equal to the random value it chooses. If it is not equal, CS_i rejects it. Otherwise, CS_i computes its authentication message $Auth_{CS_i} = h(b \cdot aP||bP||ID_{CS_i})$ and the session key between CS_i and A, $K_{CS_i-A} = h(abP||aP||bP||ID_A||ID_{CS_i})$. CS_i sends $\{ID_{CS_i}, bP, s_1, Auth_{SP}, Auth_{CS_i}\}$ to A.

Step 5. On receiving the messages from CS_i, user A computes and verifies whether $Auth_{SP} = h(K||s_1||aP)$ and $Auth_{CS_i} = h(b \cdot aP||bP||ID_{CS_i})$ hold. If one of them does not hold, A rejects them. Otherwise, A computes $K_{A-CS_i} = h(abP||aP||bP||ID_A||ID_{CS_i})$ as the session key between A and CS_i.

2.3 Password-Changing Phase

Step 1. If A wants to change the password, he/she performs the authentication phase first. After a successful authentication, A gets the secret information $h(K||s_1)$ shared with SP. Then, A inputs the new password PW_{new}, selects a random value $r' \in Z_q^*$ and submits $E_{h(K||s_1)}(ID_A, h(PW_{new}||r'))$ to SP.

Step 2. On receiving the message, SP decrypts it and obtains the new password of A. Then, SP selects another random value $R' \in \{0,1\}^{64}$ and computes $C'_A = h(s||ID_A||R') \oplus h(PW_{new}||r')$ for A. SP sends $E_{h(K||s_1)}(ID_A, C'_A)$ to A.

Step 3. A decrypts $E_{h(K||s_1)}(ID_A, C'_A)$ and updates the information in the smart card with $\{ID_A, r', C'_A\}$.

3 Weaknesses of Li et al.'s Scheme

We suppose that the adversary may intercept, insert, delete, or modify any message transmitted through the channel between the user and the server. Moreover, the secret information stored in the smart card may be exposed when the card is lost or stolen, since the secret information in it can be extracted by side channel attacks [19, 20]. Truly two-factor authentication should still be secure even either one of two factors are compromised.

Although Li et al.'s scheme is claimed to be secure against various attacks, we observe that the scheme suffers from wrong password login, DOS attack in the password-changing phase, and no provision for revocation.

3.1 Wrong Password Login

As is noted in [21], it is desired that there is an authentication test (also known as local password verification) to reject the login request if a legal user A enters a wrong password. In Li et al.'s scheme, if A mistakenly enters a wrong password, say PW'_A ($PW'_A \neq PW_A$), then Step 1 of authentication phase is still performed. Specifically, the smart card still computes $X'_A = h(PW'_A||r) \oplus C_A$ instead of $X_A = h(PW_A||r) \oplus C_A$. In this case, A will send a wrong message $\{aP, r_1, M_A, N'_A\}$ instead of the valid message $\{aP, r_1, M_A, N_A\}$. Thus, no authentication test is in place to reject wrong password, which shows the inefficiency of scheme in terms of the detection of incorrect input. This leads to unnecessarily extra communication and computational overheads during the login and authentication phase.

3.2 DOS Attack in the Password-Changing Phase

We also identify that Li et al.'s scheme suffers from DOS attack, also known as de-synchronization attack, in the password-changing phase [22]. An adversary I can mount this attack by blocking the second message in this phase. The details of the procedure are presented as follows.

Step 1. A follows the specification of password-changing phase, and submits $E_{h(K||s_1)}(ID_A, h(PW_{new}||r'))$ to SP.
Step 2. SP obtains the new password of A by decrypting the received message. Then, SP selects $R' \in \{0,1\}^{64}$ and computes $C'_A = h(s||ID_A||R') \oplus h(PW_{new}||r')$ for A. SP stores the new value R' instead of the old one R, and sends $E_{h(K||s_1)}(ID_A, C'_A)$ to A.
Step 3. I intercepts the message $E_{h(K||s_1)}(ID_A, C'_A)$. A cannot receive the message. The information in A's smart card with is still $\{ID_A, r, C_A\}$, where $C_A = h(s||ID_A||R) \oplus h(PW_A||r)$.

At this point, the value maintained by SP is the new value R', while the value stored in the smart card is $C_A = h(s||ID_A||R) \oplus h(PW_A||r)$, which is computed by the old value R. As a result, the authentication between A and SP is destined to fail when A initiates a new session to be authenticated by SP.

3.3 No Provision for Revocation

In practice, revocation of lost/stolen smart card is one of the important security demands of smart card based authentication protocols [21]. If A's smart card is lost/stolen, some mechanism is needed to prevent the misuse of lost/stolen smart card. Otherwise, an attacker can impersonate A because the registration phase is incapable to detecting the re-registration with the old identity. To address this issue, the identity table must be maintained in the server's database, through which the invalid smart card will be detected. However, most of the existing smart card based authentication schemes including Li et al.'s scheme fail to take revocation into consideration in their schemes.

4 Our Improved Scheme

To remedy these flaws presented in Sect. 3, we adopt the concept of fuzzy verifier proposed by Wang et al. [23, 24] to achieve wrong password detection and local password update. Specifically, we improve the authentication scheme of Li et al. in the following aspects. (1) The registration and authentication phase is revised to enable wrong password detection and revocation. (2) The password-changing phase is revised to avoid DOS attack. (3) Revocation and re-registration phase is added to prevent the misuse of lost/stolen smart card.

Our scheme consists of 4 phases: registration, authentication, password update, and revocation and re-registration.

4.1 Registration Phase

In the registration phase of our improve scheme, Step 1 is the same as Li et al.'s scheme. Step 2 and 3, which are different from Li et al.'s scheme, is given as follows.

Step 2. When receiving the messages, SP selects a random value $R \in \{0,1\}^{64}$ and computes $C_A = h(s||ID_A||R) \oplus h(PW_A||r)$ for A. SP updates its identity information table with the new entry $\{ID_A, R\}$, and issues a smart card which contains $\{ID_A, C_A\}$ to A.

Step 3. When receiving the smart card, A computes $HPW_A = h(h(PW_A||r) \mathrm{mod} m)$, where m is a medium integer, $2^8 \leq m \leq 2^{16}$, which determines the capacity of the pool of the PW_A against offline password guessing attack [21]. Then A stores r, HPW_A into the card. The security parameters in the smart card are $\{ID_A, r, HPW_A, C_A\}$.

4.2 Authentication and Key Exchange Phase

When A wants to access the ith cloud server CS_i, he/she needs to complete the authentication and key exchange phase. Steps 2–5 are the same as Li et al.'s scheme. Step 1, which is different from Li et al.'s scheme, is given as follows.

Step 1. $A \rightarrow CS_i$

A inserts his/her card and inputs his/her password PW_A. Then, The smart card computes $HPW'_A = h(h(PW_A||r) \bmod m)$. If the equation $HPW_A ? = HPW'_A$ does not hold, the card rejects the request. Otherwise, it continues to selects two random values $a, r_1 \in Z_q^*$. A computes $K = h(a \cdot sP)$ and $M_A = h(K||r_1) \oplus ID_A$. Next, A reveals $X_A = h(PW_A||r) \oplus C_A$, computes $N_A = h(K||r_1||X_A)$ as the authentication message, and sends $\{aP, r_1, M_A, N_A\}$ to the cloud server CS_i.

4.3 Password Change Phase

In this phase, A update the password through the following steps.

Step 1. If A needs to change his password, he inserts his card into a terminal, and enters PW_A.

Step 2. The smart card computes $HPW'_A = h(h(PW_A||r) \bmod m)$. If the equation $HPW_A ? = HPW'_A$ does not hold, the card rejects the request. Otherwise, A selects a new password PW_{new}, and calculates $C'_A = C_A \oplus h(PW_A||r) \oplus h(PW_{new}||r)$ and replace C_A with C'_A.

4.4 Revocation and Re-Registration Phase

In this phase, A can revoke his/her account and re-register without changing his/her identity ID_A.

1. For revocation of A's account, SP verifies his/her personal identities, such as personal identification card, and then simply removes the random number R from the identity information table. After the revocation of A's account, SP rejects the login request since the corresponding random value R is not presented in the identity Table
2. In the case of re-registration of A with the same identity ID_A, SP verifies whether ID_A matches with any existing entry in the identity information table. If so, SP continues to check whether the status of A is inactive. If it is true, that is, A has been already registered but the status is inactive. SP executes the registration phase to reactivate A's account. Otherwise, the re-registration request is rejected.

5 Security and Efficiency Analysis

In this section, we present the security analysis of the proposed protocol. Due to limited space, we only show that the improved protocol can resist the attacks and provide the missing features presented in Sect. 3.

5.1 Wrong Password Login Freeness

As is presented in Sect. 3.1, there is no authentication test to reject the login request when a legal user A enters a wrong password, which will lead to inefficiency and unnecessary communication and computational overheads during the login and authentication phase. However, there is an inevitable tradeoff between wrong password detection and the resistance to offline password guessing attack when the smart card is lost/stolen. In the improved scheme, we adopt the concept of fuzzy verifier proposed by Wang et al. [23, 24]. On the one hand, it can be used to provide timely wrong password and fingerprint detection when login. Specifically, the smart card checks the validity of user input password PW_A before the authentication phase. Since the smart card computes $HPW'_A = h(h(PW_A||r) \text{mod} m)$ and compares it with the stored value of HPW_A in its memory to verify the legitimacy of the user before the smart card proceeds to the following operations. On the other hand, the adversary has to perform online guessing to determine the correct password from as high as 2^12 candidates, which can be relatively easily detected and thwarted by the server by using rate-limiting and/or lockout policy [16]. Therefore, the problem of wrong password login is thwarted.

5.2 Immunity from DOS Attack in the Password-Changing Phase

In Li et al.'s scheme, if a legitimate user *A* wants to change her password, she has to send the new password in the request to *SP*, and then waits for the reply from *SP* to update the user-specific security information stored in the smart card. This interactive process enables the adversary to de-synchronize the information stored in the smart card and that maintained by *SP*. In our improved scheme, *A* can change her password without interacting with *SP*. That is, the user-specific security parameters stored in the smart card can be updated locally. Moreover, the user-specific security parameters maintained by *SP* do not need to be updated during the password-changing phase. Thus, the risk of in-consistence is eliminated, and the adversary cannot de-synchronize the information of the smart card and *SP*. Moreover, the fuzzy verifier mechanism is in place to resist wrong password entry. Therefore, DOS attack in the password-changing phase is thwarted.

5.3 Provision for Revocation and Re-Registration

In our scheme, the identity table is maintained by the service provider, through which the invalid smart card will be detected, as is presented in Sect. 4.4. As a result, revocation of lost/stolen smart card is accomplished, and the re-registration with the old identity is also detected.

5.4 Feature and Efficiency Comparison

We compare our proposed authentication scheme with existing authentication schemes [17, 25] in terms of security features. The results of comparison are shown in Table 2. From Table 2, it is obvious that our scheme is the only one which is capable of resisting all known attacks and fulfills the desirable security features.

For efficiency analysis, we compare the time complexity of our scheme and the related schemes, including the registration phase, login phase and authentication phase. The following notations are defined to facilitate the analysis.

m: the time complexity for scalar multiplication of ECC;
e: the time complexity for symmetric key encryption/decryption;
h: the time complexity of hash function.

The results of efficiency comparison are summarized in Table 3. Although the computation cost of our scheme is higher than that of [17, 25], we argue that this is because our scheme ensures the robustness of the authentication scheme and provides more security features. From Table 2, we can see that the scheme of Li et al. [25] fails to provide user anonymity and untraceability, and the scheme of Li et al. [17] cannot resist cloud server impersonation attack and denial of service attack. The additional computational cost is worthwhile in view of the security strength and features accomplished.

Table 2. Comparison of security features

Functionality\Scheme	Li et al.'s [25]	Li et al.'s [17]	Ours
Privileged insider attack	Yes	Yes	Yes
Stolen-verifier attack	Yes	Yes	Yes
Online password guessing attack	Yes	Yes	Yes
Offline password guessing attack	No	Yes	Yes
Wrong password login	No	No	Yes
Modification attack	Yes	Yes	Yes
User impersonation attack	Yes	Yes	Yes
Cloud server impersonation attack	Yes	Yes	Yes
Service provider impersonation attack	Yes	Yes	Yes
DOS attack	Yes	No	Yes
Mutual authentication	Yes	Yes	Yes
Known key security	Yes	Yes	Yes
Perfect forward secrecy	No	Yes	Yes
User anonymity	No	Yes	Yes
User untraceability	No	Yes	Yes
Provision for revocation and re-registration	No	No	Yes

Table 3. Comparison of computation cost

Schemes	Li et al.'s [36]	Li et al.'s [47]	Ours
Registration phase	$6h$	$2h$	$3h$
Authentication phase	$28h$	$6m + 4e + 14h$	$6m + 4e + 15h$
Overall computation cost	$34h$	$6m + 4e + 16h$	$6m + 4e + 18h$

6 Conclusion

Authentication is a necessity to ensure that sensitive data in the cloud are not available to illegal users. In this paper, we have used the authentication protocol of Li et al. as a case study and demonstrated the subtleties and challenges in designing a two-factor authentication and key agreement protocol. We have shown that their scheme is susceptible to the problem of wrong password login. Furthermore, their scheme is prone to DOS attack in the password-changing phase. Finally, it fails to provide user revocation when the mobile device is lost or stolen. Then, we have proposed an improved authentication and key agreement protocol to remedy these drawbacks in Li et al.'s scheme. We have shown that the proposed protocol can withstand various known attacks and provide more security features compared with Li et al.'s protocol.

Acknowledgements. This work is supported by Supported by National Natural Science Foundation of China (Program No. 61672413, U1405255, U1536202, 61372075, 61472310), National High Technology Research and Development Program (863 Program) (Program No. 2015AA016007), Natural Science Basic Research Plan in Shaanxi Province of China

(Program No. 2016JM6005), Fundamental Research Funds for the Central Universities (Program No. JB161501), and Specific project on research and development platform of Shanghai Science and Technology Committee (Program No. 14DZ2294400).

References

1. Ardagna, A., Asal, R., Damiani, E., et al.: From security to assurance in the cloud: a survey. ACM Comput. Surv. (CSUR) **48**(1), 2:1–50 (2015)
2. Li, H., Yang, Y., Luan, T., Liang, X., Zhou, L., Shen, X.: Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data. IEEE Trans. Dependable Secure Comput. **13**(3), 312–325 (2015)
3. Ren, Y., Shen, J., Wang, J., Han, J., Lee, S.: Mutual verifiable provable data auditing in public cloud storage. J. Internet Technol. **16**(2), 317–323 (2015)
4. He, D., Zeadally, S., Wu, L.: Certificateless public auditing scheme for cloud-assisted wireless body area networks. IEEE Syst. J. (2015). doi:10.1109/JSYST.2015.2428620
5. Jiang, Q., Ma, J., Li, G., Yang, L.: Robust two-factor authentication and key agreement preserving user privacy. Int. J. Netw. Secur. **16**(3), 229–240 (2014)
6. Jiang, Q., Ma, J., Lu, X., Tian, Y.: An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. Peer-to-Peer Netw. Appl. **8**(6), 1070–1081 (2015)
7. Shen, J., Tan, H., Moh, S., et al.: Enhanced secure sensor association and key management in wireless body area networks. J. Commun. Netw. **17**(5), 453–462 (2015)
8. Jiang, Q., Wei, F., Fu, S., Ma, J., Li, G., Alelaiwi, A.: Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy. Nonlinear Dyn. **83** (4), 2085–2101 (2016)
9. Fushan, W., Jianfeng, Ma., Aijun, G., Guangsong, L., Chuangui, Ma.: A provably secure three-party password authenticated key exchange protocol without using server's public-keys and symmetric cryptosystems. ITC **44**(2), 195–206 (2015)
10. Choudhury, A.J., et al.: A strong user authentication framework for cloud computing. In: Proceedings of IEEE Asia-Pacific Services Computing Conference, 12–15, pp. 110–115 (2011)
11. Hao, Z., Zhong, S., Yu, N.: A time-bound ticket-based mutual authentication scheme for cloud computing. Int. J. Comput. Commun. Control **6**(2), 227–235 (2011)
12. Pippal, R.S., Jaidhar, C.D., Tapaswi, S.: Enhanced time-bound ticket-based mutual authentication scheme for cloud computing. Informatica **37**(2), 149–156 (2013)
13. Jiang, Q., Khan, M.K., Lu, X., Ma, J., He, D.: A privacy preserving three-factor authentication protocol for e-health clouds. J. Supercomput. (2016). doi:10.1007/s11227-015-1610-x
14. Hwang, M.S., Sun, T.H.: Using smart card to achieve a single sign-on for multiple cloud services. IETE Tech. Rev. **30**(5), 410–416 (2013)
15. Tsai, J.L., Lo, N.W.: A privacy-aware authentication scheme for distributed mobile cloud computing services. IEEE Syst. J. **9**(3), 805–815 (2015)
16. Qi, J., Jianfeng, Ma., Fushan, W.: On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services. IEEE Syst. J. (2016). doi:10.1109/JSYST.2016.2574719
17. Li, H., Li, F., Song, C., et al.: Towards smart card based mutual authentication schemes in cloud computing. KSII Trans. Internet Inf. Syst. **9**(7), 2719–2735 (2015)
18. Hankerson, D., Menezes, A., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer Professional Computing. Springer, Berlin (2004)

19. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
20. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Examining smart-card security under the threat of power analysis attacks. IEEE Trans. Comput. **51**(5), 541–552 (2002)
21. Odelu, V., Das, A.K., Goswami, A.: A secure biometrics-based multi-server authentication protocol using smart cards. IEEE Trans. Inf. Forensics Secur. **10**(9), 1953–1966 (2015)
22. Jiang, Q., Ma, J., Li, G., et al.: An efficient ticket based authentication protocol with unlinkability for wireless access networks. Wireless Pers. Commun. **77**(2), 1489–1506 (2014)
23. Wang, D., He, D., Wang, P., Chu, C.H.: Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. IEEE Trans. Dependable Secure Comput. **12** (4), 428–442 (2015)
24. Wang, D., Wang, P.: On the usability of two-factor authentication. In: Tian, J., Jing, J., Srivatsa, M. (eds.) International Conference on Security and Privacy in Communication Networks. LNICS, vol. 152, pp. 141–150. Springer, Heidelberg (2014)
25. Li, X., Xiong, Y., Ma, J., Wang, W.: An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. J. Netw. Comput. Appl. **35**(2), 763–769 (2012)