Secure and Efficient Two-Factor Authentication Protocol Using RSA Signature for Multi-server Environments

Zhiqiang Xu[1], Debiao He[1(✉)], and Xinyi Huang[2]

[1] State Key Lab of Software Engineering, Computer School, Wuhan University, Wuhan, China
jkebxzq@163.com, hedebiao@163.com

[2] Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, Fuzhou, China
xyhuang81@yahoo.com

Abstract. To avoid multiple number of registrations using multiple passwords and smart-cards, many two-factor multi-server authentication protocols based on RSA have been proposed. However, most of the existing RSA-based multi-server authentication protocols are susceptible to various security attacks, and have high computation complexities. Recently, Amin et al. proposed a two-factor RSA-based robust authentication system for multi-server environments. However, we found that Amin et al.'s protocol cannot resist common modulus attack. To enhance the security, we propose a secure two-factor RSA-based authentication protocol for multi-server environments. The performance and security features of our scheme are also compared with that of the similar existing schemes. The performance and security analysis show that our protocol achieves more security features and has lower computation complexity in comparison with the latest related schemes.

Keywords: RSA · Smart card · User authentication
Multi-server environment

1 Introduction

User authentication scheme is essential for implementing the secure communication because it can provide mutual authentication. Two-factor authentication protocols are widely used to ensure secure communication between the remote client and the server. It is a critical task to design a secure and robust two-factor authentication protocols. To ensure the security of client-server communication in single server environment, many authentication protocols using RSA cryptosystem [1–3], hash function [1,4,5], chaotic map [6,7] and elliptic curve [8] have been proposed. However, most of these protocols cannot be used in multi-server environments. To address the issue, many authentication protocols for

S. Qing et al. (Eds.): ICICS 2017, LNCS 10631, pp. 595–605, 2018.
https://doi.org/10.1007/978-3-319-89500-0_51

multiple server environments have been designed. The multi-server communications contain three entities: the registration center, the users and multiple application-servers. All the users and application-servers must register themselves to the registration center. The responsibility of application-servers is to provide remote services for the users.

One of the most important aspects in authentication protocol is user anonymity [9,10]. The adversary shouldn't obtain user's identity in many application areas such as wireless sensor network [11], medical system and banking, where the adversary cannot guess or extract user's identity in the phase of login and authentication.

1.1 Related Work

To achieve strong security as well as lower complexities, the researchers have designed a lot of authentication protocols for multiple servers environments. Liao et al. [12] designed a authentication protocol for multi-server environments. Unfortunately, Hsiang et al. [13] showed that Liao et al.'s protocol cannot withstand several common attacks, and then they improved the identified weaknesses. However, Lee et al. [14] demonstrated that Hsiang et al.'s scheme also have some security issues. To solve these problems, they designed an efficient and enhanced authentication protocol. Unfortunately, Troung et al. [15] demonstrated that their scheme cannot withstand user impersonation and smart-card stolen attacks. To overcome these problems, they designed a secure authentication protocol in [15]. Further, Sood et al. [16] demonstrated that Hsiang et al.'s protocol [13] had different security weaknesses and put forwarded a new authentication protocol. Subsequently, Li et al. [17] showed that Sood et al.'s protocol cannot resist smart-card stolen attack and proposed a new protocol.

Recently, Pippal et al. [18] put forward a new user authentication protocol and claimed that their protocol could resist known attacks. However, He et al. [19] showed that their protocol cannot withstand impersonation attacks. In 2015, Giri et al. [2] put forward a secure protocol based on RSA and showed that their protocol could resist known attacks. Unfortunately, Amin and Biswas [1] pointed out that Giri et al.'s protocol could not against several security attacks, and then they devised a new protocol. However, Arshad et al. [20] demonstrated that [1] was not secure and proposed a new RSA-based authentication protocol.

1.2 Our Contributions

We put forward a secure and efficient two-factor authentication protocol based on RSA signature. Our major contributions are summarized as follows:

- Firstly, we analyse Amin et al.'s scheme [21] and prove it cannot withstand the common modulus attack.
- Secondly, we put forward a secure and efficient two-factor authentication protocol using RSA signature for multi-server environment. The new scheme can resist against various attacks.

– Finally, we analyse the security of our scheme and demonstrate that it is provably secure. Moreover, the performance analysis shows that our scheme is better in terms of communication overheads and computation.

1.3 Organization of the Article

The organization of this paper are described as follows: Sect. 2 analyzes the security problems of [21] and then a secure and efficient two-factor authentication protocol using RSA signature for multi-server environments are proposed in Sect. 3. Section 4 elaborates the security of the new scheme briefly. Furthermore, the comparison with some relevant protocols for efficiency and security are presented Sect. 5. Finally, we draw a conclusion.

2 Security Analysis of Amin et al.'s Scheme

In this section, we analyze the security of [21]. From our analysis, their scheme is insecure against common modulus attack. We present a list of symbols used throughout this article in Table 1. The details are described as follows:

Any of the application server can extract a public key e_{j2} and it's public/private key pair (e_{j1}, d_{j1}). All the application servers share the public modulu $\phi(n)$, where $n = p * q$, $\phi(n) = (p-1)(q-1)$. It can computes $e_{j1}d_{j1} \equiv 1 \bmod \phi(n)$, $e_{j1}d_{j1} - 1 = k\phi(n)$, $\phi(n) \mid e_{j1}d_{j1} - 1$. It can computes $(e_{j1}d_{j1} - 1, e_{j2}) = s$ using euclidean algorithm. If $s = 1$, there exist some f, g such that $f(e_{j1}d_{j1} - 1) + ge_{j2} = 1$. So the value of g is the private key corresponding to e_{j2}. If $s \neq 1$, suppose that $t = e_{j1}d_{j1} - 1/s$, $(e_{j1}d_{j1} - 1/s, e_{j2}) = 1$. There exist some f, g such that $ft + ge_{j2} = 1$ using extended euclidean algorithm. Therefore, $ge_{j2} \equiv 1 \bmod \phi(n)$, the value of g is the private key corresponding to e_{j2}.

Suppose that the adversary dispatches the same message m whose encryption exponents respectively are e_{j1} and e_{j2}. Suppose further that $gcd(e_{j1}, e_{j2}) = 1$,

Table 1. Notations

Symbol	Meaning
U_i	User
AS_j	Application-server
RC	Registration center
ID_i	Identity of U_i
ID_j	Identity of AS_j
g	A generator g $\in Z_n^*$
e	Public key of RC
d	Private key of RC
a, r	Random number selected by the U_i in authentication phase

it can computes $c_1 = m^{e_{j1}}$ mod n, $c_2 = m^{e_{j2}}$ mod n. There exist some r, s such that $re_{j1}+se_{j2} = 1$, $m = m^{re_{j1}+se_{j2}} = (m^{e_{j1}})^r (m^{e_{j2}})^s = c_1^r c_2^s$ mod n. Therefore, the adversary can get the value of m using c_1, c_2, r, s.

3 Proposed Protocol

We put forward a secure authentication system for multi-server environment which can withstand the above mentioned security issues. Our scheme consists of three phases: application-server registration phase, user registration phase, verification phase.

3.1 Application-Server Registration Phase

- AS_j selects two large prime numbers p, q, and computes $n_j = p_j \times q_j$, $\phi(n_j) = (p_j - 1)(q_j - 1)$.
- AS_j chooses a public key e_j $(1 < e_j < \phi(n_j))$, where $gcd(\phi(n_j), e_j) = 1$. Then, it computes it's private key $d_j \equiv e_j^{-1} \bmod \phi(n_j)$.
- Finally, AS_j chooses his/her identity ID_j and sends $\langle e_j, n_j, ID_j \rangle$ to RC securely.
- RC computes $Cer_j = h(e_j \,||\, ID_j \,||\, n_j)^d$ and sends Cer_j to AS_j.

3.2 User Registration Phase

- U_i chooses his/her identity ID_i and sends it to RC.
- After receiving $\langle ID_i \rangle$, RC computes $d_i = h(ID_i)^d \bmod n_j$.
- RC sends d_i to U_i securely.

3.3 Authentication Phase

- U_i randomly selects a number T_i, then it sends T_i to AS_j.
- After receiving T_i, AS_j computes $A_j = h(T_i)^{d_j}$. Then, it sends $\langle e_j, n_j, Cer_j, A_j \rangle$ to U_i.
- Upon receiving the message, U_i checks whether $Cer_j^e \bmod n_j = h(ID_j \,||\, e_j \,||\, n_j)$ and $A_j^{e_j} = h(T_i)$. If holds, U_i authenticates AS_j.
- U_i chooses three random number a_i, r, m and computes $PID_i = (ID_i \oplus a_i \,||\, a_i)^{e_j} \bmod n_j$, $R_i = h(PID_i)^r \bmod n_j$, $x = h(m, R_i)$ and $S_i = d_i^{r-x}$. Then, U_i sends $\langle PID_i, R_i, S_i, x \rangle$ to AS_j.
- After receiving the message, AS_j computes $S_i^{e_j} = h(ID_i)^{r-x}$, $PID_i^{d_j} \bmod n_j = ID_i \oplus a_i \,||\, a_i$, $ID_i^{'} = ID_i \oplus a_i \,||\, a_i$. Then, it checks whether $S_i^{e_j} h\left(ID_i^{'}\right)^x = R_i$. If holds, AS_j authenticates U_i.

4 Security Analysis

4.1 Security Proof

In this subsection, according to the formal security model described as [22], we show our protocol is secure as follows.

Theorem 1. If has advantage $Adv_P^{ake}(A)$ against our scheme running in time q_{send} Send queries, q_{exe} Execute queries and q_h Hash queries. Define the security length l and the password space $|D|$. Then, we can attain: $Adv_P^{ake} \leq \frac{q_h^2}{2^{l-1}} + \frac{(q_{send}+q_{exe})^2}{p} + 2q_h \cdot Adv_G^{DLP}(t) + \frac{q_{send}}{2^{l-2}} + \frac{2q_{send}}{|D|}$, where $Adv_G^{DLP}(t)$ denote the probabilistic polynomial time t to breach DLP problem.

Proof: C obtains (y, g, n) and intends to compute x satisfying $g^x = y \bmod n$ using the PPT turingmachine A. C utilizes the hash function as a random oracle and maintains an empty $H - list$. We define G_i as the sequence of games and Suc_i as A gets b successfully.

Game G_0: This game corresponds to the real attack, we have $Adv_P^{ake}(A) = 2Pr[Suc_0] - 1$.

Game G_1: In this game, we imitate the hash oracles $h(\cdot)$ by maintaining hash list L_h and the Execute, Reveal, Send, Corrupt and Test oracles are simulated as real attacks (see Figs. 1 and 2). Therefore, we have $Pr[Suc_1] = Pr[Suc_0]$.

Game G_2: In this game, we imitate all oracles as previous games, except that we will halt all executions under the condition: A collision occurs in the transcript $\langle e_j, n_j, Cer_j, A_j \rangle$, $\langle PID_i, R_i, S_i, x \rangle$. According to the birthday paradox, the probability of the hash oracle collisions is $\frac{q_h^2}{2^{l+1}}$. The probability of the transcripts colisions is $\frac{(q_{send}+q_{exe})^2}{2p}$.

Game G_3: In this game, we simulate all oracles as previous games, except that we will cancel all executions where in the adversary guess the authentication parameters A_j and R_i without making hash query. Therefore, we have $Pr[Suc_3] - Pr[Suc_2] \leq \frac{q_{send}}{2^l}$.

Game G_4: In this game, we imitate all oracles under the condition that the adversary guess the parameter $h(T_i)$ successfully without making the related queries. We define $k = h(T_i)$ to imitate this game.

- AS_j: Search for $(*, k)$ in L_h. This game will be terminated if the information does not exist. Otherwise, compute $A_j = k^{d_j}$.
- U_i: Computes $A_j^{e_j}$ and checks whether $A_j^{e_j} = h(T_i)$. If holds, U_i search for $\langle e_j, n_j, Cer_j, A_j \rangle$ in the send list.

If A guess the parameter k successfully without making hash quries, this game will succeed. Therefore, we have $Pr[Suc_3] - Pr[Suc_2] \leq \frac{q_{send}}{2^l}$.

Game G_5: We design this game to imitate Discrete logarithm problem. The security of our protocol depends on the Discrete logarithm problem solely: $R_i = h(PID_i)^r \bmod n^j$.

On hash oracle queries C maintains a hash list L_h. The tuple $\{x, y\}$ is in L_h and $y = h(x)$. After receiving the queries from A, the response from C is as follows:

- Quries $\{x, y\}$ in L_h and returns y if it exists.
- Otherwise, selects a number y and responds it to A. Finally, adds $\{x, y\}$ into L_h.

On a query Send(U_i, *start*), assuming U_i is in the correct state, U_i responds the query as follows:

- Chooses a nonce T_i.
- Responds the information $\{T_i\}$.

On a query Send($AS_j, \{T_i\}$), assuming AS_j is in the prospective state, AS_j responds the query as follows:

- Calculates $A_j = h(T_i)^{d_j}$.
- Responds the information $\{Cer_j, e_j, n_j, A_j\}$.

On a query Send($U_i, \{Cer_j, e_j, n_j, A_j\}$), assuming U_i is in the correct state, U_i responds the query as follows:
Checks whether $Cer_j^e \bmod n_j = h(ID_j \,||\, e_j \,||\, n_j)$ and $A_j^{e_j} = h(T_i)$.

- Terminates this session if not holds.
- Otherwise, chooses two random number a_i, r. Then computes: $PID_i = (ID_i \oplus a_i \,||\, a_i)^{e_j} \bmod n_j$, $R_i = h(PID_i)^r \bmod n_j$, $x = h(m, R_i)$, $S_i = d_i^{r-x}$.
- Responds the message $\langle PID_i, R_i, S_i, x\rangle$.

On a query Send($AS_j, \{PID_i, R_i, S_i, x\}$), Assuming AS_j is in the prospective state, AS_j responds the query as follows:
Calculates $S_i^{e_j} = h(ID_i)^{r-x}$, $PID_i^{d_j} \bmod n_j = ID_i \oplus a_i \,||\, a_i$, $ID_i' = ID_i \oplus a_i \,||\, a_i$.
Then checks whether $S_i^{e_j} h\left(ID_i'\right)^x = R_i$.

- Aborts the message $\langle PID_i, R_i, S_i, x\rangle$ if not holds.
- Otherwise, the message $\langle PID_i, R_i, S_i, x\rangle$ is accepted.

Fig. 1. Simulation of Send query.

- U_i: Chooses two random numbers a_i and r. Then calculates: $PID_i = (ID_i \oplus a_i \,||\, a_i)^{e_j} \bmod n_j$, $S_i = d_i^{r-x}$. Finally, stores $\{S_i, x\}$ into hash list.
- AS_j: Calculates $S_i^{e_j} = h(ID_i)^{r-x}$, $PID_i^{d_j} \bmod n_j = ID_i \oplus a_i \,||\, a_i$, $ID_i' = ID_i \oplus a_i \,||\, a_i$. Then stores $\left\{S_i^{e_j}, h(ID_i)^x\right\}$ into hash list.

This game is different from previous games where in the adversary quries $h(\cdot)$ on $g^x = y \bmod n$. We can obtain DLP secret with $\frac{1}{q_h}$. Hence, we have $Pr[Suc_5] - Pr[Suc_4] \leq q_h \cdot Adv_G^{DLP}(t)$.

Game G_6: In this game, we simulate all oracles as in game G_5, except that we will abort the Test query in which A asks a $h(\cdot)$ for $g^x = y \bmod n$. The probability that A gets the session key is $\frac{q_h^2}{2^{l+1}}$. Therefore, we have $Pr[Suc_5] - Pr[Suc_4] \leq \frac{q_h^2}{2^{l+1}}$.

In addition, the probability of off-line dictionary attacks is $\frac{q_{send}}{|D|}$. Therefore, we can get the conclusion showed in the beginning of this subsection.

On a query Execute, we proceed using the simulation of the Send query as follows:

- $\{T_i\} \leftarrow \text{Send}(U_i, start)$.
- $\{Cer_j, e_j, n_j, A_j\} \leftarrow \text{Send}(AS_j, \{T_i\})$.
- $\langle PID_i, R_i, S_i, x\rangle \leftarrow \text{Send}(U_i, \{Cer_j, e_j, n_j, A_j\})$

Finally, the query responds $\{T_i\}$, $\{Cer_j, e_j, n_j, A_j\}$ and $\langle PID_i, R_i, S_i, x\rangle$.

On a query Corrupt, we proceed as follows:

- If $a = 1$, it outputs U_i's password.
- Otherwise, it outputs the secret parameters of U_i stored in smart card.

On a query Reveal, we proceed as follows:

- If $\prod_{i,j}^{n}$ has accepted, it responds the session key betweeen $\prod_{i,j}^{n}$ and its partner.
- Else it outputs a null value.

On a query Test, we proceed as follows: It flips a fair coin b.

- If $b = 1$, it returns the right parameters.
- Otherwise, it responds a random value with the same size.

Fig. 2. Simulation of Execute, Reveal, Test query.

4.2 Other Discussions

This subsection shows our scheme is able to withstand various attacks.

- User impersonation attack: To impersonate as a legal U_i, A has to compute a valid message $\langle PID_i, R_i, S_i, x\rangle$ during authentication phase, where $PID_i = (ID_i \oplus a_i \,||\, a_i) \bmod n_j$, $R_i = h(PID_i)^r \bmod n_j$, $x = h(m, R_i)$ and $S_i = d_i^{r-x}$. However, it is infeasible to compute PID_i and R_i without knowing the random number a_i and r. Therefore, our proposed scheme can withstand user impersonation attack.
- Server impersonation attack: To impersonate as a legal application-server, A has to compute the message $\langle e_j, n_j, Cer_j, A_j\rangle$, which is to be authenticated by U_i, where $A_j = h(T_i)^{d_j}$ and $Cer_j = h(e_j \,||\, ID_j \,||\, n_j)^d$. However, it is infeasible to compute A_j and Cer_j without knowing the private key d of RC. Therefore, our proposed scheme can withstand server impersonation attack.
- User anonymity: Our proposed scheme can provide anonymity of users. Taking the situation where an adversary can get the message d_i, where $d_i = h(ID_i)^d \bmod n_j$. However, d cannot be known by the adversary. The adversary may also eavesdrop the information $\langle PID_i, R_i, S_i, x\rangle$, where PID_i are related to the user's identity and $PID_i = (ID_i \oplus a_i \,||\, a_i)^{d_j}$. As the random a_i cannot be known by the adversary, it is impossible to get the user's identity from PID_i. Therefore, our proposed scheme can provide anonymity of users.
- Common modulus attacks: In our proposed scheme, the public key and private key of AS_j are generated by the server itself. AS_j computes $n_j = p_j \times q_j$, $\phi(n_j) = (p_j - 1)(q_j - 1)$. The modulu of n_j is not same in every application server and the application servers does not share the public modulu. Therefore, our proposed scheme can withstand Common modulus attack.

- Mutual authentication: In our proposed scheme, AS_j and U_i authenticate each other. U_i authenticates AS_j by checking whether $Cer_j^e \bmod n_j = h(ID_j \,||\, e_j \,||\, n_j)$ and $A_j^{e_j} = h(T_i)$. A needs to get T_i to reconstruct A_j, however, only a legal AS_j owns the value. AS_j authenticates U_i by checking whether $S_i^{e_j} h\left(ID_i^{'}\right)^x = R_i$. A needs to compute PID_i and r to reconstruct R_i, however, only a legal U_i can compute those values. Therefore, U_i and AS_j mutually authenticate and our proposed scheme can provide proper mutual authentication.
- Smart-card stolen attack: An adversary A can extract the information of smart-card by means of power consumption monitoring technique. Suppose that A obtains the smart card of U_i and extracts the information $\langle d_i \rangle$, where $d_i = h(ID_i)^{d_j} \bmod n_j$. From the value, A cannot compute any useful information, because the value is safeguarded with a one-way hash function. Further, A cannot obtain the value of d_j. Therefore, our proposed scheme can withstand smart card stolen attacks.

5 Performance Analysis

In this section, we compare the proposed scheme with recent authentication schemes [18,21,23–25] proposed in terms of security and performance (as shown in Table 2). We use some time complexities to evaluate the computational cost. T_h denotes the cost time for one-way hash operation. T_{sym} denotes the execution time for symmetric key encryption/decryption operation. T_e denotes the running time for exponentiation operation. T_m denotes the execution time for modular multiplication operation.

We have implemented various cryptographic operations with the MIRACL C/C++ Library [26] on a personal computer with 4G bytes memory and the Windows 7 operating system. It requires Visual C++ 2008, 1024-bit cyclic group, AES for symmetric encryption/decryption, 160-bit prime field F_p and

Table 2. Comparison of security

Security attributes and schemes	[18]	[23]	[24]	[25]	[21]	Our scheme
User anonymity	√	√	√	√	√	√
Stolen smart card attack	√	√	√	√	√	√
Impersonation attack	×	√	√	√	√	√
Replay attack	√	√	√	√	√	√
Denial of service attack	√	√	√	√	√	√
Session key verification	×	×	×	×	√	√
Man-in-the-middle	×	√	√	√	√	√
Common modulus attacks	√	√	√	√	×	√

Table 3. Comparison of computation cost at the user side and the server side

Schemes	User	Server
Pippal et al. [18]	$4T_h + 3T_e + T_m$	$3T_h + 4T_e + T_m$
Yeh et al. [23]	$4T_h + 2T_e + T_m$	$5T_h + 4T_e + T_m$
Wei et al. [24]	$7T_h + 2T_e$	$6T_h + 2T_e$
Li et al. [25]	$5T_h + T_e$	$8T_h + 3T_e$
Amin et al. [21]	$6T_h + 2T_e$	$6T_h + 2T_e + T_{sym}$
Proposed scheme	$5T_h + 2T_e$	$5T_h + 2T_e$

Table 4. Comparison of execution time at the user side and the server side

Schemes	User	Server
Pippal et al. [18]	5.4966 ms	7.3235 ms
Yeh et al. [23]	3.6701 ms	7.5621 ms
Wei et al. [24]	3.6566 ms	3.6562 ms
Li et al. [25]	1.8289 ms	5.4839 ms
Amin et al. [21]	3.6562 ms	3.7865 ms
Proposed scheme	3.5861 ms	3.6132 ms

SHA-1 operation. The execution time of these different operations are: 0.0004 ms, 0.1303 ms, 0.0147 ms and 1.8269 ms.

In Table 1, we find that our proposed scheme can withstand known attacks, such as user anonymity, common modulus attacks, mutual authentication, Server impersonation attack. In Tables 3 and 4, we find that computational cost time of our proposed scheme is lower than the schemes in [18,23] and nearly equal with the schemes in [21,24,25].

6 Conclusion

In this paper, we cryptanalyzed Amin et al.'s scheme, and found that their protocol is susceptible to common modulus attack. Then We present a secure and efficient two-factor authentication protocol using RSA signature for multi-server environments. We prove informally that our protocol can withstand different cryptographic attacks. In the proposed scheme, we employ RSA signature to implement the authentication scheme. Our proposed scheme is suitable for deployment in various low-power smart cards, and in particular for the mobile computing networks.

Acknowledgements. The work of was supported by the National Natural Science Foundation of China (Nos. 61501333, 61572379, 61572370, 61772377), and the Natural Science Foundation of Hubei Province of China (Nos. 2015CFA068, 2017CFA007).

References

1. Amin, R., Biswas, G.P.: An improved RSA based user authentication and session key agreement protocol usable in TMIS. J. Med. Syst. **39**(8), 1–14 (2015)
2. Giri, D., Maitra, T., Amin, R., Srivastava, P.D.: An efficient and robust RSA-based remote user authentication for telecare medical information systems. J. Med. Syst. **39**(1), 1–9 (2015)
3. Amin, R., Biswas, G.P.: Remote access control mechanism using rabin public key cryptosystem. In: Mandal, J.K., Satapathy, S.C., Sanyal, M.K., Sarkar, P.P., Mukhopadhyay, A. (eds.) Information Systems Design and Intelligent Applications. AISC, vol. 339, pp. 525–533. Springer, New Delhi (2015). https://doi.org/10.1007/978-81-322-2250-7_52
4. Amin, R., Biswas, G.P.: Cryptanalysis and design of a three-party authenticated key exchange protocol using smart card. Arab. J. Sci. Eng. **40**(11), 1–15 (2015)
5. Islam, S.K.H., Biswas, G.P., Choo, K.K.R.: Cryptanalysis of an improved smartcard-based remote password authentication scheme. Inf. Sci. Lett. **3**(1), 35 (2014)
6. Hafizul Islam, S.K., Khan, M.K., Obaidat, M.S., Bin Muhaya, F.T.: Provably secure and anonymous password authentication protocol for roaming service in global mobility networks using extended chaotic maps. Wirel. Pers. Commun. **84**(3), 1–22 (2015)
7. Hafizul Islam, S.K.: Design and analysis of a three party password-based authenticated key exchange protocol using extended chaotic maps. Inf. Sci. Int. J. **312**(C), 104–130 (2015)
8. Amin, R., Biswas, G.P.: A secure three-factor user authentication and key agreement protocol for TMIS with user anonymity. J. Med. Syst. **39**(8), 1–19 (2015)
9. Hafizul Islam, S.K.: Design and analysis of an improved smartcard-based remote user password authentication scheme. Int. J. Commun. Syst. **29**(11), 1708–1719 (2016)
10. Hafizul Islam, S.K.: A provably secure id-based mutual authentication and key agreement scheme for mobile multi-server environment without ESL attack. Wirel. Pers. Commun. **79**(3), 1975–1991 (2014)
11. Amin, R., Biswas, G.P.: A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. Ad Hoc Netw. **36**, 58–80 (2016)
12. Liao, Y.-P., Wang, S.-S.: A secure dynamic ID based remote user authentication scheme for multi-server environment. Comput. Stand. Interfaces **31**(1), 24–29 (2009)
13. Hsiang, H.-C., Shih, W.-K.: Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. Comput. Stand. Interfaces **31**(6), 1118–1123 (2009)
14. Lee, C.-C., Lin, T.-H., Chang, R.-X.: A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards. Expert Syst. Appl. **38**(11), 13863–13870 (2011)
15. Truong, T.-T., Tran, M.-T., Duong, A.-D.: Robust secure dynamic ID based remote user authentication scheme for multi-server environment. In: Murgante, B., Misra, S., Carlini, M., Torre, C.M., Nguyen, H.-Q., Taniar, D., Apduhan, B.O., Gervasi, O. (eds.) ICCSA 2013. LNCS, vol. 7975, pp. 502–515. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39640-3_37

16. Sood, S.K., Sarje, A.K., Singh, K.: A secure dynamic identity based authentication protocol for multi-server architecture. J. Netw. Comput. Appl. **34**(2), 609–618 (2011)
17. 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)
18. Pippal, R.S., Jaidhar, C.D., Tapaswi, S.: Robust smart card authentication scheme for multi-server architecture. Wirel. Pers. Commun. **72**(1), 729–745 (2013)
19. He, D., Chen, J., Shi, W., Khan, M.K.: On the security of an authentication scheme for multi-server architecture. Int. J. Electr. Secur. Digit. Forensics **5**(3/4), 288–296 (2013)
20. Arshad, H., Rasoolzadegan, A.: Design of a secure authentication and key agreement scheme preserving user privacy usable in telecare medicine information systems. J. Med. Syst. **40**(11), 237 (2016)
21. Amin, R., Islam, S.K., Khan, M.K., et al.: A two-factor RSA-based robust authentication system for multiserver environments. Secur. Commun. Netw. **2017**, 15 p. (2017). Article no. 5989151
22. Ding, W., Ping, W.: Two birds with one stone: two-factor authentication with security beyond conventional bound. IEEE Trans. Dependable Secure Comput. **PP**(99), 1 (2016)
23. Yeh, K.-H.: A provably secure multi-server based authentication scheme. Wirel. Pers. Commun. **79**(3), 1621–1634 (2014)
24. Wei, J., Liu, W., Hu, X.: Cryptanalysis and improvement of a robust smart card authentication scheme for multi-server architecture. Wirel. Pers. Commun. **77**(3), 2255–2269 (2014)
25. Li, X., Niu, J., Kumari, S., Liao, J., Liang, W.: An enhancement of a smart card authentication scheme for multi-server architecture. Wirel. Pers. Commun. Int. J. **80**(1), 175–192 (2015)
26. Lili, X., Fan, W.: Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care. J. Med. Syst. **39**(2), 10 (2015)