Hindawi
Security and Communication Networks
Volume 2017, Article ID 5989151, 15 pages
https://doi.org/10.1155/2017/5989151

Research Article

A Two-Factor RSA-Based Robust Authentication System for Multiserver Environments

Ruhul Amin,[1] SK Hafizul Islam,[2] Muhammad Khurram Khan,[3] Arijit Karati,[4] Debasis Giri,[5] and Saru Kumari[6]

[1]*Computer Science and Engineering, Thapar University, Patiala, Punjab 147004, India*
[2]*Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India*
[3]*Center of Excellence in Information Assurance, King Saud University, Riyadh 11451, Saudi Arabia*
[4]*Computer Science and Engineering, NIIT University, Neemrana, Rajasthan 301705, India*
[5]*Computer Science and Engineering, Haldia Institute of Technology, Haldia, West Bengal 721657, India*
[6]*Department of Mathematics, Chaudhary Charan Singh University, Meerut, Uttar Pradesh 250004, India*

Correspondence should be addressed to SK Hafizul Islam; hafi786@gmail.com

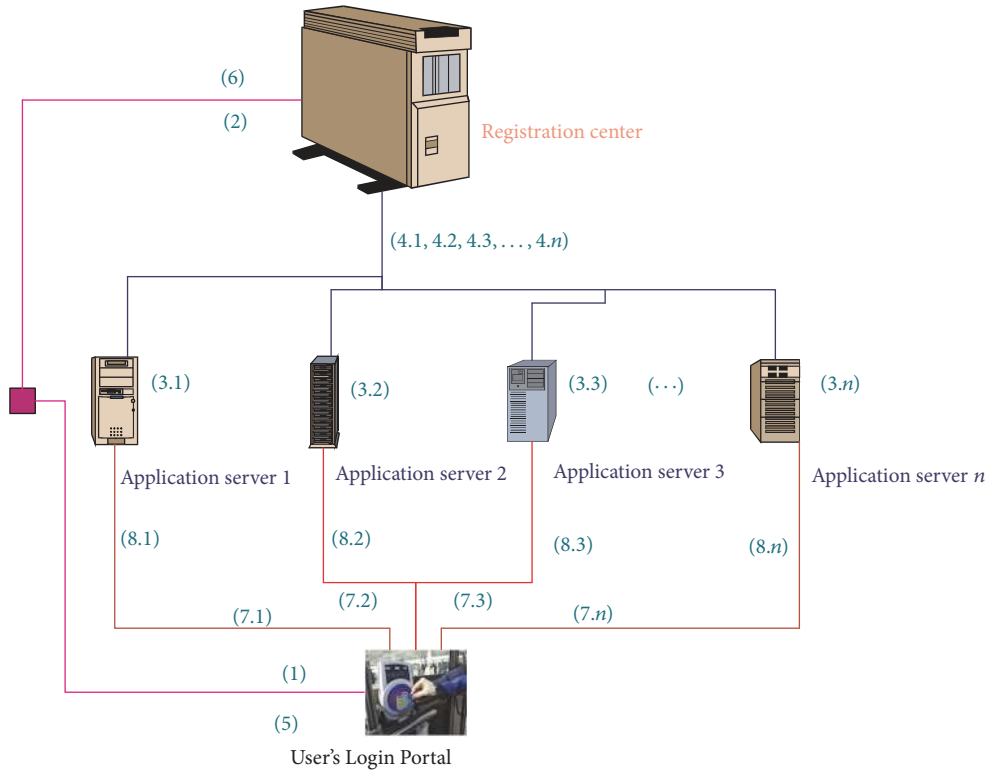
Received 4 April 2017; Accepted 18 May 2017; Published 21 August 2017

Academic Editor: Sherali Zeadally

The concept of two-factor multiserver authentication protocol was developed to avoid multiple number of registrations using multiple smart-cards and passwords. Recently, a variety of two-factor multiserver authentication protocols have been developed. It is observed that the existing RSA-based multiserver authentication protocols are not suitable in terms of computation complexities and security attacks. To provide lower complexities and security resilience against known attacks, this article proposes a two-factor (password and smart-card) user authentication protocol with the RSA cryptosystem for multiserver environments. The comprehensive security discussion proved that the known security attacks are eliminated in our protocol. Besides, our protocol supports session key agreement and mutual authentication between the application server and the user. We analyze the proof of correctness of the mutual authentication and freshness of session key using the BAN logic model. The experimental outcomes obtained through simulation of the Automated Validation of Internet Security Protocols and Applications (AVISPA) S/W show that our protocol is secured. We consider the computation, communication, and storage costs and the comparative explanations show that our protocol is flexible and efficient compared with protocols. In addition, our protocol offers security resilience against known attacks and provides lower computation complexities than existing protocols. Additionally, the protocol offers password change facility to the authorized user.

1. Introduction

Nowadays, network technologies and the online applications are rapidly growing; the Internet users are accessing online services to make their life more comfortable over Internet. When accessing a remote server, a session key should be negotiated between user and remote server, which would be used to encrypt the confidential messages. In this regard, two-factor authentication protocols using smart-card and password are widely used and designing a robust and efficient two-factor authentication protocol is a critical task. The concept of client-server communication over insecure networks has been introduced for single server environment, and many researchers have proposed numerous authentication protocols [1–3] using hash function [4–6], RSA cryptosystem [7–9], elliptic curve [10], chaotic map [11, 12], and bilinear pairing [13]. However, these protocols are unsuitable for multiserver communications. For this purpose, lots of authentication protocols [13–17] have been put forward for multiserver architecture. In multiserver communications, there are three entities such as a user, a registration center, and a number of application servers. In this environment, all the application servers and users registered themselves to the registration center. All of these application servers are responsible for providing required services to the users. An architecture of the multiserver communication is given in Figure 1.



(1): user's registration request
(2): reply from registration center
(3.1, 3.2, . . . , 3.n): registration request from application server(s)
(4.1, 4.2, 4.3, . . . , 4.n): reply from registration center
(5): login request
(6): reply against login message
(7.1, 7.2, 7.3, . . . , 7.n): mutual authentication
(8.1, 8.2, 8.3, . . . , 8.n): session key agreement

Figure 1: Multiserver framework used in our protocol.

User anonymity [18, 19] is one of the security issues in user authentication protocol, where user's identity should be kept secret from the adversary. There are several application areas such as banking, medical system [7, 10], and wireless sensor network [20], where the authorized users want to keep their identity secret from an adversary; that is, the adversary is unable to extract or guess user's original identity from the login and authentication messages. In addition to that, the mutual authentication is another important security aspect, where client and server both can verify each other to avoid man-in-the-middle attack. As the adversary can also act as server, so verifying at the server end is also important. Like mutual authentication, protection of low-entropy information such as identity and password from guessing attack is very imperative. Furthermore, the insider attack is also difficult to protect since a person who is managing the registration center may misuse users' confidential information such as password. So, the protection of such kind of attack is also essential. As we know the public messages of any authentication protocol are transmitted via open channels, and thus an adversary may intercept, drop, modify, and reroute the captured messages. Thus, the adversary can try to impersonate either the user or the remote server. In this regard, mutual authentication [18, 19] between the participants is most important security aspect. Another objective of any client-server authentication protocol is to negotiate a session key between entities. Since both the participants negotiate their session key, whether the session keys computed by the both entities are equal or not should be checked. In order to perform this, a session key verification property [21] should be involved in the protocol.

1.1. Literature Reviews. In the literature, numerous multiserver authentication protocols are designed to achieve strong security as well as lower complexities. In 2009, Liao and wang [22] put forward a multiserver authentication protocol; however, Chen et al. [23] showed that the protocol in [22] suffers from forward secrecy problem. Further, Hsiang and Shih [24] demonstrated that the protocol in [22] cannot defy some common attacks [13], and then they enhanced it to remedy the identified weaknesses. Lee et al. [25] designed an enhanced and efficient multiserver authentication protocol after pointing out the security weakness of the protocol in [24]. Subsequently, Troung et al. [26] illustrated that the protocol in [24] cannot resist smart-card stolen and user impersonation attacks. To vanquish these loopholes, the authors put forward another multiserver authentication protocol and showed that it is secure against common attacks. Moreover,

Sood et al. [15] observed different security weaknesses of the protocol in [24] and then designed a multiserver authentication protocol. In 2012, Li et al. [17] showed that the protocol in [15] cannot withstand smart-card stolen attack and subsequently put forward a new protocol. Currently, Pippal et al. [14] designed a secure smart-card based user authentication protocol and showed that their protocol is robust against known attacks. Unfortunately, He et al. [27] demonstrated that the protocol in [14] is not secure against impersonation attack, offline password guessing attack, insider attack, and server spoofing attack. Also, Yeh [16] demonstrated that the protocol in [14] is not secure against some common attacks. More recently, Hsieh and Leu [28] demonstrated that Liao and Hsiao's protocol [29] could not give full security requirements and then devised an efficient protocol using bilinear pairing. In 2015, Amin and Biswas [13] showed that the protocol in [28] is insecure against offline password guessing attack, server masquerading attack, and user anonymity problem. Additionally, they described that the password change facility of this protocol is not user-friendly. To overcome these security vulnerabilities, Amin and Biswas [13] devised user authentication and session key agreement protocol using smart-card and bilinear pairing. In the same year, Giri et al. [8] proposed a robust authentication protocol using RSA and claimed that it is strongly protected against security vulnerabilities. But, Amin and Biswas [7] highlighted that the protocol in [8] is not secure against several security attacks. Further, they proposed a new protocol to overcome the security weaknesses. In 2016, Sutrala et al. [30] show that the protocol in [7] is not secure and also proposed an improved protocol. Similarly, the authors in [31] demonstrated security attacks of the protocol in [7] and proposed a new RSA-based authentication protocol.

1.2. Contributions. This article claims the following contributions.

(i) We have pointed out that all the existing RSA-based multiserver protocols include high computation cost and are insecure against known attacks.

(ii) The main contribution of this article is the design and analysis of a lightweight and robust authentication protocol for multiserver environments.

(iii) The proposed protocol is analyzed using the AVISPA S/W and its results showed that our protocol is secure.

(iv) We proved that our protocol is completely protected against known security threats. Further, it solves the problem of anonymity issue.

(v) We show that the performance of our protocol is better in terms of computation and communication overheads.

1.3. Organization of the Article. Section 2 discusses our authentication protocol for multiserver architecture. The concrete security discussion of our protocol appears in Section 3. The authentication correctness of our protocol using BAN logic is shown in Section 4. The simulation results of our protocol using AVISPA S/W are discussed in Section 5. The comprehensive performance measurement of our protocol and comparative results with other protocols appear in Section 6. We have provided the conclusion with some remarks and future scope in Section 7.

2. Proposed Authentication Protocol

Based on RSA cryptosystem, this section introduces a two-factor multiserver authentication protocol using password and smart-card. Our authentication protocol includes initialization phase, registration phase for application server AS_j and user U_i, login phase, verification phase, and password change phase. We present a list of symbols used throughout this article in the Notations.

2.1. Initialization Phase. This phase is executed by the RC to initialize the whole system. Accordingly, RC determines two large prime numbers p and q (each of them at least 1024 bits), computes $n = p \times q$, and further determines a generator $g \in Z_n^*$. RC keeps (p, q) as secrets and publishes n as public value. RC further selects a cryptographic hash function $h(\cdot)$: $\{0,1\}^* \rightarrow \{0,1\}^l$, where $\{0,1\}^*$ represents arbitrary binary string and $\{0,1\}^l$ is a fixed-length l binary string.

2.2. Application Server Registration Phase. The main objective of AS_j is to provide remote services to U_i on demand through insecure communication. In order to provide remote services, all AS_j must execute registration phase to the trusted RC. During registration, RC and AS_j perform the operations as given below.

Step 1. AS_j first selects his/her identity ID_j and submits it to RC through a secure channel.

Step 2. After receiving the registration request, RC chooses a public key e_j $(1 < e_j < \phi(n))$ such that $\gcd(\phi(n), e_j) = 1$ and computes the corresponding private key $d_j \equiv e_j^{-1} \bmod \phi(n)$, where $\phi(n) = (p-1)(q-1)$. RC further computes confidential information $Y_j = h(e_j \parallel d_j)$.

Step 3. RC maintains a table RCT that includes $\langle ID_j, e_j, (Y_j \oplus x)\rangle$ and keeps it securely. Finally, RC sends d_j to AS_j securely.

We further depict the registration phase of application server of the proposed protocol in Figure 2.

2.3. User Registration Phase. To register with RC, U_i performs the tasks as explained below.

Step 1. U_i determines his/her identity ID_i together with password PW_i and sends $\langle ID_i, A_i\rangle$ to RC over a secure channel, where $A_i = h(ID_i \parallel PW_i)$.

Step 2. After receiving $\langle ID_i, A_i\rangle$, RC computes $B_i = g^{x \cdot ID_i} \bmod n$, $C_i = ID_i^{A_i} \bmod n$, $D_i = B_i \oplus C_i$, $E_i = h(A_i \parallel B_i)$, and $F_i = ENC_x(ID_i)$ and then embeds $\langle D_i, E_i, F_i, h(\cdot), n\rangle$ into an new and fresh smart-card.

Step 3. After preparing the smart-card, RC sends it to U_i over secure channel or in person.

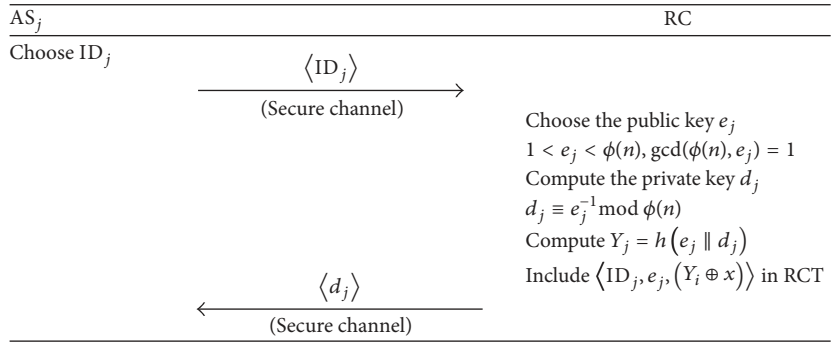


Figure 2: Application server registration phase of our protocol.

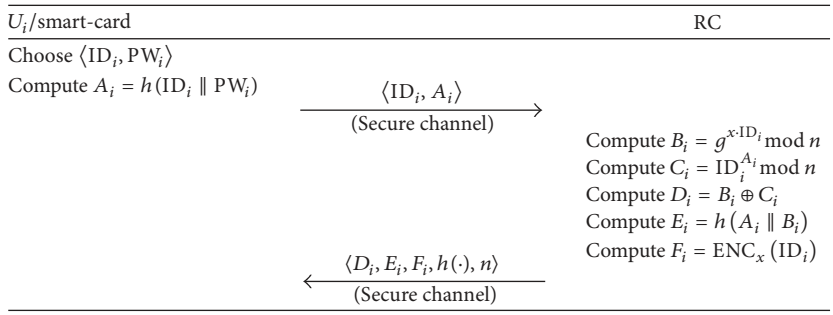


Figure 3: User registration phase of our protocol.

Further we demonstrate the user registration phase of our protocol in Figure 3.

2.4. Login Phase. At the time of enjoying remote services, U_i takes initiation and executes this phase. The following operations are performed in this phase.

Step 1. U_i inserts the smart-card and enters $\langle \mathrm{ID}_i, \mathrm{PW}_i \rangle$ to the terminal on request. Then, the terminal computes $A_i^* = h(\mathrm{ID}_i^* \parallel \mathrm{PW}_i^*)$, $C_i^* = \mathrm{ID}_i^{A_i^*} \bmod n$, $B_i^* = D_i \oplus C_i^*$, and $E_i^* = h(A_i^* \parallel B_i^*)$ and checks the condition whether E_i^* matches with E_i. If the condition holds, it implies that $(\mathrm{ID}_i^* = \mathrm{ID}_i)$ and $(\mathrm{PW}_i^* = \mathrm{PW}_i)$.

Step 2. The terminal now produces a random number R_i and calculates $M_1 = h(B_i^* \parallel R_i)$ and $M_2 = h(B_i^*) \oplus R_i$ and then sends $\langle M_1, M_2, F_i, \mathrm{ID}_j \rangle$ to RC using an insecure channel.

Further we provide description of login phase of the proposed protocol in Figure 4.

2.5. Verification Phase. This phase achieves a session key agreement between U_i and AS_j with the help of RC over insecure communication. All the operations performed by RC, U_i, and AS_j are discussed now.

Step 1. After getting login message $\langle M_1, M_2, F_i, \mathrm{ID}_j \rangle$, RC first decrypts F_i using its own secret key x and then further computes $B_i' = g^{x \cdot \mathrm{ID}_i} \bmod n$, $R_i' = M_2 \oplus h(B_i')$, and $M_1' = h(B_i' \parallel R_i')$. Then RC checks whether M_1' matches with M_1. If the condition holds, U_i is verified by RC.

Step 2. RC extracts Y_j and the public key e_j and then computes $M_3 = Y_j \oplus \mathrm{ID}_i$. Finally, RC sends $\langle e_j, M_3 \rangle$ to U_i via insecure channel.

Step 3. After receiving $\langle e_j, M_3 \rangle$, U_i computes $Y_j' = M_3 \oplus \mathrm{ID}_i$, $M_4 = h(\mathrm{ID}_i \parallel R_i \parallel Y_j' \parallel \mathrm{ID}_j)$, and $M_5 = (\mathrm{ID}_i \parallel R_i \parallel M_4)^{e_j} \bmod n$ and then transmits M_5 to AS_j over insecure channel.

Step 4. After receiving M_5, AS_j first decrypts M_5 using its private key d_j and according to RSA cryptosystem, AS_j obtains $\langle \mathrm{ID}_i, R_i, M_4 \rangle$. Then AS_j computes $Y_j'' = h(e_j \parallel d_j)$ and $M_4'' = h(\mathrm{ID}_i \parallel R_i \parallel Y_j'' \parallel \mathrm{ID}_j)$ and then checks whether M_4'' matches with M_4. If such condition is incorrect, AS_j terminates the connection or else authenticates U_i.

Step 5. After verifying U_i, AS_j computes the session key $\mathrm{SK}_j = h(\mathrm{ID}_i \parallel \mathrm{ID}_j \parallel R_i \parallel R_j)$ and further computes $R_{ij} = R_i \oplus R_j$ and $M_6 = h(\mathrm{SK}_j \parallel R_j \parallel \mathrm{ID}_j)$. AS_j sends $\langle M_6, R_{ij} \rangle$ to U_i through any insecure channel.

Step 6. After receiving $\langle M_6, R_{ij} \rangle$, U_i computes $R_j' = R_{ij} \oplus R_i$, $\mathrm{SK}_i = h(\mathrm{ID}_i \parallel \mathrm{ID}_j \parallel R_i \parallel R_j')$, and $M_6' = h(\mathrm{SK}_i \parallel R_j' \parallel \mathrm{ID}_j)$ and checks whether $M_6' =? M_6$ holds. If the condition holds, U_i authenticates AS_j and session key generated by U_i and AS_j is verified.

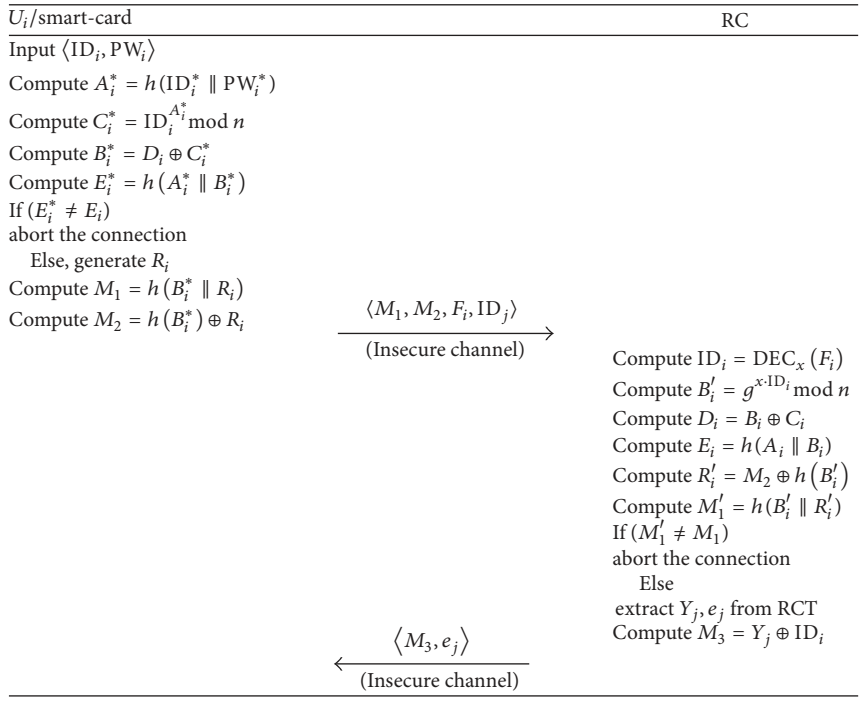


Figure 4: Login phase of our protocol.

Further we provide the description of verification phase of our protocol in Figure 5.

2.6. Password Change Phase. In password-based remote user authentication protocol, user's password is one of the most sensitive information pieces and must be changed whenever required. Therefore, password change phase must be rendered to the registered user. This phase requires the following actions to change the password.

Step 1. U_i inserts the smart-card and enters old $\langle \mathrm{ID}_i, \mathrm{PW}_i \rangle$ to the terminal on request. Then, the terminal computes $A_i^* = h(\mathrm{ID}_i^* \parallel \mathrm{PW}_i^*)$, $C_i^* = \mathrm{ID}_i^{A_i^*} \bmod n$, $B_i^* = D_i \oplus C_i^*$, and $E_i^* = h(A_i^* \parallel B_i^*)$ and checks whether E_i^* matches with E_i. If the condition is correct, it implies that $(\mathrm{ID}_i^* = \mathrm{ID}_i)$ and $(\mathrm{PW}_i^* = \mathrm{PW}_i)$.

Step 2. After successfully verifying $\langle \mathrm{ID}_i, \mathrm{PW}_i \rangle$, the smart-card made request to the U_i for a new password. Then U_i enters a fresh password PW_i^n and then the smart-card computes $A_i^n = h(\mathrm{ID}_i \parallel \mathrm{PW}_i^n)$, $C_i^n = \mathrm{ID}_i^{A_i} \bmod n$, $D_i^n = B_i^* \oplus C_i^n$, and $E_i^n = h(A_i^n \parallel B_i^*)$ and replaces $\langle D_i, E_i \rangle$ with $\langle D_i^n, E_i^n \rangle$.

Further we explained the our password change phase in Figure 6.

3. Discussion about the Security

This section measures security strength of our protocol by analyzing several security attacks. Generally, password-based authentication protocol suffers from different attacks. The authentication protocol must resist the known attacks and achieve mutual authentication, session key verification along with login, and password change phase efficiently. Here, we analyze that our protocol can defy all the relevant security attacks. In this regard, we define some valid assumptions [2, 10, 20] as follows.

Definition 1. An adversary $\mathscr{A}$ has the capabilities for retrieving smart-card information using power consumption monitoring method [34, 35]. For instance, if $\mathscr{A}$ gets the smart-card, he/she has the capability of finding the confidential data from it.

Definition 2. All messages produced by the protocol during execution are sent via insecure communication such as Internet. Therefore, $\mathscr{A}$ intercepts, deletes, modifies, reroutes, and resends the transmitted message. It is worth noting that $\mathscr{A}$ cannot intercept any information from the secure network [10].

Definition 3. $\mathscr{A}$ knows execution of the protocol; that is, the protocol is public. Additionally, $\mathscr{A}$ may behave like legitimate user or client.

Definition 4. We considered that the authorized user selects dictionary word as password and identity. It is noted that $\mathscr{A}$ is unable to guess password and identity in polynomial time. If $A_i = h(\mathrm{ID}_i \parallel \mathrm{PW}_i)$ is known to $\mathscr{A}$, then it is hard to verify guessed password and identity using A_i in polynomial time [20].

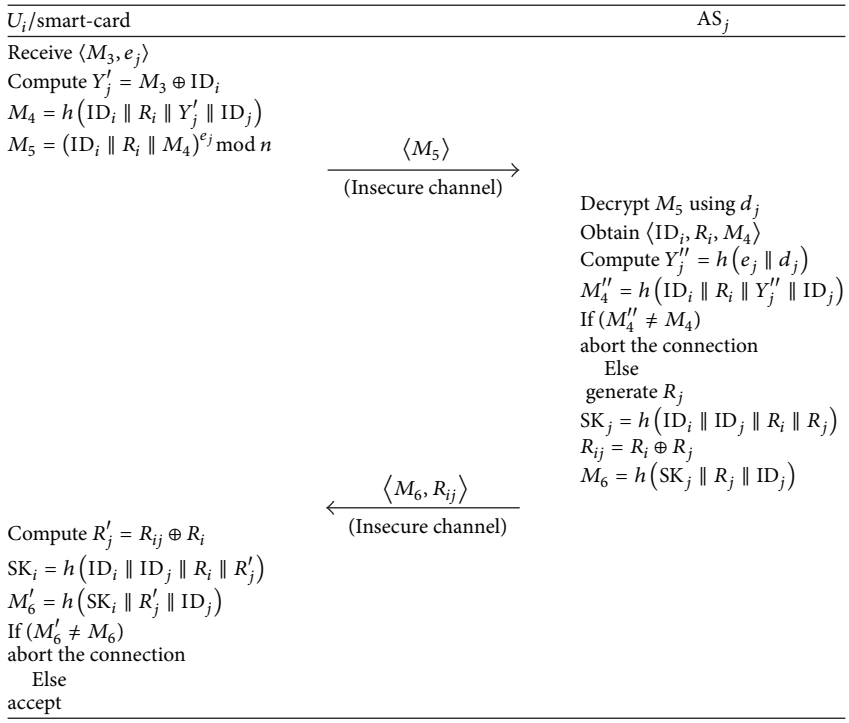


Figure 5: Verification phase of our protocol.

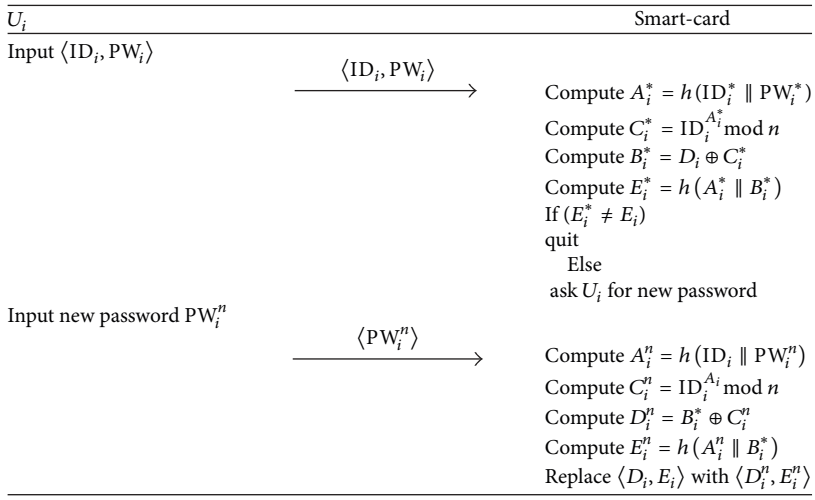


Figure 6: Password update phase of our protocol.

Definition 5. We considered that the random nonce and secret key size are significantly large (parameter with high entropy). Therefore, $\mathcal{A}$ is unable to guess the information within polynomial time.

Definition 6. Let $A_i = B_i \oplus C_i$ and only A_i is known to $\mathcal{A}$. Then $\mathcal{A}$ cannot find B_i or C_i from A_i within polynomial time.

Definition 7. Assume length of identity or password is n characters. According to [36], the probability of guessing the password is approximately equal to $1/2^{6n}$.

Theorem 8. *Our multiserver authentication protocol resists smart-card stolen attack.*

Proof. In smart-card stolen attack, $\mathcal{A}$ generally extracts all the smart-card information (see Definition 1) and then tries to login to AS_j on behalf of U_i. Suppose $\mathcal{A}$ got smart-card of U_i and extracted the information $\langle D_i, E_i, F_i, h(\cdot), n \rangle$, where $B_i = g^{x \cdot ID_i} \bmod n$, $C_i = ID_i^{A_i} \bmod n$, $D_i = B_i \oplus C_i$, $E_i = h(A_i \| B_i)$, and $F_i = ENC_x(ID_i)$. Note that $\mathcal{A}$ cannot compute any confidential information of U_i or AS_j using D_i. Further, E_i and F_i are protected by the hash function and symmetric key

encryption algorithm, respectively. Therefore, the extraction of secure information by $\mathcal{A}$ is infeasible and the protocol resists smart-card stolen attack. □

Theorem 9. *Our multiserver authentication protocol can defy the offline identity guessing attack.*

Proof. As per Theorem 8, it is clear that $\mathcal{A}$ has no advantage to found confidential information from the lost/stolen smart-card of any legal user. In this attack, we further elaborated whether $\mathcal{A}$ can guess user's identity using intercepted public messages. We supposed that $\mathcal{A}$ intercepts the login messages $\langle M_1, M_2, F_i, \mathrm{ID}_j\rangle$, $\langle e_j, M_3\rangle$, and $\langle M_5, M_6, R_{ij}\rangle$, where $M_1 = h(B_i^* \parallel R_i)$, $M_2 = h(B_i^*) \oplus R_i$, $M_3 = Y_j \oplus \mathrm{ID}_i$, $M_4 = h(\mathrm{ID}_i \parallel R_i \oplus Y_j' \parallel \mathrm{ID}_j)$, $M_5 = (\mathrm{ID}_i \parallel R_i \parallel M_4)^{e_j} \bmod n$, $R_{ij} = R_i \oplus R_j$, and $M_6 = h(\mathrm{SK}_j \parallel R_j \parallel \mathrm{ID}_j)$. Note that $\mathcal{A}$ cannot guess ID_i using B_i without knowing the secret key d of RC. Further, $\mathcal{A}$ cannot guess ID_i from M_3 without Y_j. If $\mathcal{A}$ wants to verify the guessed identity using B_i^* and M_3, the probability would be approximately $1/2^{6n+1024}$ and $1/2^{6n+128}$, respectively, where length of d is 1024 bits. The value of M_5 is transmitted as ciphertext, and therefore, it is also infeasible to extract information without the private key d_j of AS_j. Hence, the proposed authentication protocol can defy this type of offline identity guessing attack. □

Theorem 10. *Our multiserver authentication protocol can defy the offline password guessing attack.*

Proof. Similar to Theorem 9, $\mathcal{A}$ may attempt to guess PW_i of U_i in offline mode. We described in Theorem 8 that $\mathcal{A}$ cannot extract or guess U_i's password PW_i from the smart-card information. In addition, the public messages are also independent of U_i's PW_i. As a result, $\mathcal{A}$ has no way to guess U_i's password in offline mode. □

Theorem 11. *Our multiserver authentication protocol can defy the insider attack.*

Proof. As mentioned in [7, 10], if the insider person $\mathcal{A}$ of RC or AS_j gets U_i's password PW_i by some means, she/he can attempt to login different accounts of U_i. Thus, the protection against the insider attack is also an important issue in password-based authentication system. When the registration phase of our protocol is executed, U_i delivers ID_i and A_i to RC for issuing a smart-card, where $A_i = h(\mathrm{ID}_i \parallel \mathrm{PW}_i)$. Since A_i is protected by hash function, $\mathcal{A}$ is unable to extract PW_i due to one-way behavior of the hash function. □

Theorem 12. *Our multiserver authentication protocol can defy the user impersonation attack.*

Proof. To impersonate U_i, $\mathcal{A}$ has to generate a valid login message that must be authenticated by RC. During the login phase, U_i generates $\langle M_1, M_2, F_i, \mathrm{ID}_j\rangle$, where $M_1 = h(B_i^* \parallel R_i)$ and $M_2 = h(B_i^*) \oplus R_i$. Note that $\mathcal{A}$ can generate the random number R_i. However, it is very difficult to compute $B_i = g^{x\cdot \mathrm{ID}_i} \bmod n$ without knowing $\langle x, \mathrm{ID}_i\rangle$ from the known information. Therefore, the protocol resists user impersonation attack. □

Theorem 13. *Our multiserver authentication protocol can defy the server impersonation attack.*

Proof. Resembling Theorem 12, $\mathcal{A}$ may attempt to forge the authentication message to AS_j. For that, $\mathcal{A}$ has to compute another genuine message $\langle M_6, R_{ij}\rangle$, which is to be authenticated by U_i, where $R_{ij} = R_i \oplus R_j$ and $M_6 = h(\mathrm{SK}_j \parallel R_j \parallel \mathrm{ID}_j)$. Now, $\mathcal{A}$ generates the random number $R_j' (\neq R_j)$ and tries to compute the valid R_{ij}. But, it is infeasible to compute R_{ij} without knowing R_i. On the other hand, $\mathcal{A}$ cannot compute correct M_6 without knowing SK_j. Hence, the adversary is unable to launch this attack. □

Theorem 14. *Our multiserver authentication protocol can defy the session key computation attack.*

Proof. The protocol negotiates a session key between U_i and AS_j, which will be used to encrypt the confidential information transmitted over public channel. Our authentication protocol computes the session key $\mathrm{SK}_j = \mathrm{SK}_i = h(\mathrm{ID}_i \parallel \mathrm{ID}_j \parallel R_i \parallel R_j)$ and its security depends on the hash function. Note that session key depends on the confidential information $\langle \mathrm{ID}_i, R_i, R_j\rangle$. We discussed earlier that $\mathcal{A}$ is unable to compute these pieces of information. Hence, $\mathcal{A}$ is unable to calculate the session key of our protocol. □

Theorem 15. *Our multiserver authentication protocol can defy the known session specific temporary information attack.*

Proof. It is assumed that short-term information used in the protocol is known to $\mathcal{A}$. However, $\mathcal{A}$ could not compute the session key in any session. We assume that $\mathcal{A}$ knows R_i and R_j tries to compute SK_j. Then $\mathcal{A}$ needs ID_i to compute SK_j. We described earlier that the computation of ID_i is not feasible by $\mathcal{A}$. Therefore, $\mathcal{A}$ cannot launch this attack. □

Theorem 16. *Our multiserver authentication protocol can defy the stolen verifier attack.*

Proof. At the time of application server registration, RC maintains a table called RCT, which includes $\langle \mathrm{ID}_j, e_j, (Y_j \oplus x)\rangle$. Among these, Y_j is the most confidential parameter used in the proposed protocol. We now supposed that $\mathcal{A}$ got RCT by some means and attempted to break the security of the proposed protocol. Note that Y_j is not stored in plaintext form. It is stored as $Y_j \oplus x$ in the table, where x is the secret key of RC. As a result, $\mathcal{A}$ cannot compute Y_j without knowing x. □

4. Authentication Proof Using BAN Logic

This section addresses the freshness of the exchanged message and the proof of the mutual authentication based on Burrows-Abadi-Needham (BAN) logic [9, 13]. Being a known security model, BAN logic is required to demonstrate the authentication and secure key distribution protocols. Now,

Table 1: Logical postulates of BAN logic.

Rule	Notation	Description
Message meaning	$\frac{P \mid\equiv P \overset{K}{\rightleftharpoons} Q, P \triangleleft \langle X \rangle_K}{P \mid\equiv Q \mid\sim X}$	If principal P believes K is shared with Q and sees $\langle X \rangle_K$, then P believes Q once said X.
Freshness-conjuncatenation	$\frac{P \mid\equiv \sharp(X)}{P \mid\equiv \sharp(X, Y)}$	If P believes X is fresh, then P believes the freshness of (X, Y).
Belief	$\frac{P \mid\equiv (X), P \mid\equiv Y}{P \mid\equiv (X, Y)}$	If principal P believes X and Y, then P believes (X, Y).
Nonce verification	$\frac{P \mid\equiv \sharp(X, P \mid\equiv Q \mid\sim X)}{P \mid\equiv Q \mid\equiv X}$	If P believes X is fresh and the principal Q once sent X, then P believes that Q believes X.
Jurisdiction	$\frac{P \mid\equiv Q \Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$	If the principal believes that Q has jurisdiction over X and Q believes X, then P believes that X is true.
Session key	$\frac{P \mid\equiv \sharp(X), P \mid\equiv Q \mid\equiv X}{P \mid\equiv P \overset{K}{\leftrightarrow} Q}$	If P believes session key is fresh and P and Q believe X, which are the necessary parameters of session key, then P believes that (s)he shares session key K with Q.

some backgrounds and notations used in BAN logic are given below.

(i) *Principals* are agents involved in the protocol (usually people or programs).

(ii) *Keys* are required to encrypt messages symmetrically.

(iii) *Public Keys* are similar to Keys except they are used in pairs.

(iv) *Nonces* are message parts that are not repeated.

(v) *Timestamps* are similar to nonces where they are unlikely to be repeated.

The BAN statements used in this paper that are required to analyze the security of our protocol are listed in the Notations.

The logical postulates of the BAN logic are listed in Table 1.

In order to prove the authentication protocol is valid, a list of the following things must be processed:

(i) In language of formal logic, idealize our protocol.

(ii) Find out assumptions about the initial state of our protocol.

(iii) In order to deduce new predicates, use production and rules of the logic.

(iv) Use logic to find out beliefs made by parties in our protocol.

To show our protocol is secure, it must satisfy BAN logic based goals as discussed below.

(i) *Goal 1*: $U_i \mid\equiv U_i \overset{\mathrm{SK}}{\longleftrightarrow} \mathrm{AS}_j$

(ii) *Goal 2*: $U_i \mid\equiv \mathrm{AS}_j \mid\equiv U_i \overset{\mathrm{SK}}{\longleftrightarrow} \mathrm{AS}_j$

(iii) *Goal 3*: $\mathrm{AS}_j \mid\equiv \mathrm{AS}_j \overset{\mathrm{SK}}{\longleftrightarrow} U_i$

(iv) *Goal 4*: $\mathrm{AS}_j \mid\equiv U_i \mid\equiv \mathrm{AS}_j \overset{\mathrm{SK}}{\longleftrightarrow} U_i$

Our protocol is transformed into the idealized form:

$$M_1: U_i \rightarrow \mathrm{AS}_j : M_5,\ \langle R_i \rangle_{Y_j}$$

$$M_2: AS_j \rightarrow U_i : M_6,\ \langle R_j \rangle_{\mathrm{ID}_i}$$

The assumptions on initial state of the protocol are made to analyze the proposed protocol as follows:

$$A_1: U_i \mid\equiv \sharp(R_i)$$

$$A_2: \mathrm{AS}_j \mid\equiv \sharp(R_j)$$

$$C_1: U_i \mid\equiv U_i \overset{Y_j}{\longleftrightarrow} \mathrm{AS}_j$$

$$C_2: \mathrm{AS}_j \mid\equiv \mathrm{AS}_j \overset{\mathrm{ID}_i}{\longleftrightarrow} U_i$$

$$B_1: \mathrm{AS}_j \mid\equiv U_i \Rightarrow R_i$$

$$B_2: U_i \mid\equiv \mathrm{AS}_j \Rightarrow R_j$$

Idealized form of the protocol is demonstrated under BAN logic with assumptions. The main proofs are as follows:

$M_1: U_i \rightarrow \mathrm{AS}_j : M_5,\ \langle R_i \rangle_{Y_j}$.

Based on seeing rule, we get

$S1$: $\mathrm{AS}_j \triangleleft M_5$

According to C_2, $S1$, and message meaning rule, we know

$S2$: $\mathrm{AS}_j \mid\equiv U_i \mid\sim R_i$.

According to A_2, $S2$, and freshness-conjuncatenation rule and nonce verification rule, we know

$S3$: $\mathrm{AS}_j \mid\equiv U_i \mid\equiv R_i$, where R_i is the essential session key parameter of the proposed protocol.

According to B_1, $S3$, and the jurisdiction rule is applied, we know

$S4$: $\mathrm{AS}_j \mid\equiv R_i$.

According to A_2, $S3$, and the session key rule is used, we get

$S5$: $\mathrm{AS}_j \mid\equiv \mathrm{AS}_j \overset{\mathrm{SK}}{\longleftrightarrow} U_i$ *(Goal 3)*.

$S5$ with nonce verification rule according to A_2 is used and we get

$S6$: $\mathrm{AS}_j \mid\equiv U_i \mid\equiv \mathrm{AS}_j \overset{\mathrm{SK}}{\longleftrightarrow} U_i$ *(Goal 4).*

M_2: $\mathrm{AS}_j \rightarrow U_i : M_6,\ \langle R_j \rangle_{\mathrm{ID}_i}$.

Based on seeing rule it gets

$S7$: $U_i \triangleleft M_6$.

Based on C_1, $S7$, and message meaning rules, it is known that

$S8$: $U_i \mid\equiv \mathrm{AS}_j \mid\sim R_j$.

Based on A_1, $S8$, and nonce verification and freshness-conjuncatenation rules, we get

$S9$: $U_i \mid\equiv \mathrm{AS}_j \mid\equiv R_j$, where R_j is essential session key parameter of our protocol.

Based on B_2, $S9$, and jurisdiction rule, we get

$S10$: $U_i \mid\equiv R_j$.

Based on A_1, $S9$, and the session key rule, we get

$S11$: $U_i \mid\equiv U_i \overset{\mathrm{SK}}{\longleftrightarrow} \mathrm{AS}_j$ *(Goal 1).*

$S11$ and nonce verification rule are used according to A_1 and get

$S12$: $U_i \mid\equiv \mathrm{AS}_j \mid\equiv U_i \overset{\mathrm{SK}}{\longleftrightarrow} \mathrm{AS}_j$ *(Goal 2).*

The discussion made above proves that our protocol correctly achieves the mutual authentication property.

5. Simulation Results of the Scheme Using AVISPA

We first briefly discuss the concept of the AVISPA and then present simulation results followed by HLPSL code of our protocol. According to literature review, the AVISPA is one of the widely used simulation tools for verifying security correction of the authentication protocol. Many earlier protocols [7, 10, 20] have been simulated using AVISPA tool.

5.1. Description of the AVISPA Tool. It is well known that AVISPA [37] is known as a tool for formal security verification. AVISPA supports HLPSL (High Level Scheme Specification Language). A general description and pictorial representation of the AVISPA tool can be found in [20, 37, 38]. AVISPA tool supports four different back-ends (OFMC, CL-AtSe, SATMC, and TA4SP) and abstraction based methods that are integrated through HLPSL. We refer to [38] for more details of OFMC, CL-AtSe, SATMC, and TA4SP. During the execution of the tool, HLPSL specification is interpreted into intermediate form (IF) by hlpsl2if translator where IF is lower-level language and directly read by AVISPA back-ends.

The HLPSL is a role-oriented language where every party plays a role during protocol execution. Every role is free from others, retrieving some initial information through parameters and communicating with other roles by the channels. Intruder is framed using the model [39] with a possibility for intruder as a legitimate role in protocol run. In addition, role system narrates principals, number of sessions, and roles. OUTPUT FORMAT (OF) using the four back-ends is produced, and (OF) after successful execution highlights the result whether our protocol is safe or not under which condition output is generated. Based on the HLPSL language, we have implemented the role specifications for the user, registration center, and application server in Algorithms 1, 2, and 3, respectively. The role specification for the session, goal, and environment in HLPSL is given in Algorithm 4.

```
role user (Ui, RC, ASj: agent,
SKey1 : symmetric_key,
SKey2 : symmetric_key,
H, MUL, SUB: hash_func,
Snd, Rcv: channel(dy))
played_by Ui
def=
local State : nat,
IDi, PWi, IDj, N, Ej, Yj: text,
Di, Ei, Fi, Ai, Bi, Ci, Ri, M1, M2, M3, M4, M6, Rij, Rj, M5,
SKi: message,
Inc : hash_func
const user_rserver, rserver_aserver,
aserver_user, sec1, sec2, sec3, sec4, sec5, sec6: protocol_id
init State :=0
transition
1. State = 0 ∧ Rcv(start) =|>
State' := 1 ∧ Ai' := h(IDi.PWi)
∧ Snd({IDi.Ai'}_SKey2)
2. State = 1 ∧ Rcv({Di'.Ei'.Fi'.N'}_SKey2) =|>
State' := 2 ∧ Ci' := exp(IDi, Ai)
∧ Bi' := xor(Di',Ci')
∧ Ri' :=new()
∧ M1' :=h(Bi'.Ri')
∧ M2' := xor(h(Bi'), Ri')
∧ Snd(M1'.M2'.Fi.IDj)
∧ secret({PWi}, sec1, {Ui})
3. State = 2 ∧ Rcv(Ej'.M3') =|>
State' := 3 ∧ Yj' := xor(M3',IDi)
∧ M4' := h(IDi. Ri. Yj'.IDj)
∧ M5' :=exp((IDi.Ri.M4'), Ej)
∧ Snd(M5')
∧ secret({Yj'}, sec2, {Ui,ASj,RC})
∧ witness(ASj, Ui, aserver_user, Ri)
4. State = 3 ∧ Rcv(M6',Rij') =|>
State' := 4 ∧ Rj' := xor(Rij,Ri)
∧ SKi' := h(IDi.IDj.Ri.Rj')
∧ secret({SKi}, sec3, {Ui,ASj})
end role
```

Algorithm 1: Role specification for the user of our protocol in HLPSL.

5.2. Simulation Results. Algorithms 5 and 6 show the results on back-ends CL-AtSe and OFMC. Results for both the back-ends confirm that the protocol is SAFE which indicates that the protocol resists passive and active attacks. The proposed protocol achieves the following six properties and two authentication phases that are mentioned below.

(1) Secrecy_of sec 1: PW_i is the confidential information and known to U_i only.

```
role rserver (RC, Ui, ASj: agent,
SKey1 : symmetric_key,
SKey2 : symmetric_key,
H, MUL, SUB: hash_func,
Snd, Rcv: channel(dy) )
played_by RC
def=
local State : nat,
IDj, IDi, Ej, P, Q, N, W, Dj, G, Yj: text,
Ai, Bi, Ci, Di, Ei, Fi, X, M1, M2, M3: message,
Inc : hash_func
const user_rserver, rserver_aserver,
aserver_user, sec1, sec2, sec3, sec4, sec5,
sec6 : protocol_id
init State :=0
transition
1. State = 0 ∧ Rcv({IDj'}_SKey1) =|>
State' := 1 ∧ Ej' := new()
∧ Dj' :=new()
∧ P' :=new()
∧ Q' :=new()
∧N' := MUL(P',Q')
∧W' := MUL(SUB(P',1), SUB(Q',1))
∧ Yj' := h(Ej'.Dj')
∧ Snd({Ej'.Dj'}_SKey1)
∧ secret({Dj'}, sec4, {ASj,RC})
∧ secret({P',Q'}, sec5, {RC})
2. State = 1 ∧ Rcv({IDi.Ai'}_SKey2) =|>
State' := 2 ∧ Bi' := exp(G,MUL(X,IDi))
∧ Ci' := exp(IDi,Ai)
∧ Di' := xor(Bi',Ci')
∧ Ei' :=h(Ai.Bi')
∧ Fi' := {IDi}_X
∧ Snd({Di'.Ei'.Fi'.N}_SKey2)
3. State = 2 ∧ Rcv(M1'.M2'.Fi.IDj) =|>
State' := 3 ∧ M3' := xor(Yj,IDi)
∧ Snd(Ej.M3')
end role
```

Algorithm 2: Role specification for registration server of our protocol in HLPSL.

(2) Secrecy_of sec 2: the confidential information Y_j is known to $\langle U_i, \mathrm{AS}_j, \mathrm{RC}\rangle$ only.

(3) Secrecy_of sec 3: the session key SK_i is shared only between U_i and AS_j.

(4) Secrecy_of sec 4: the private key d_j of AS_j is shared between $\langle \mathrm{AS}_j, \mathrm{RC}\rangle$.

(5) Secrecy_of sec 5: the confidential information P and Q pieces are only known to RC.

(6) secrecy_of sec 6: the random numbers $\langle R_i, R_j\rangle$ are known to $\langle \mathrm{AS}_j, U_i\rangle$ only.

(7) authentication_on alice_server_Ri: U_i produces a random nonce R_i, and if AS_j gets it secretly, then AS_j authenticates U_i.

(8) authentication_on server_alice_Rj: U_i authenticates AS_j based on the random number R_j.

```
role aserver (ASj, Ui, RC: agent,
SKey1 : symmetric_key,
SKey2 : symmetric_key,
% H is hash function
H,MUL,SUB: hash_func,
Snd, Rcv: channel(dy) )
played_by ASj
def=
local State : nat,
IDi, IDj, Ej, Dj, Yj: text,
M5, M9, Ri, M4, Rj, Rij, SKj, M6: message,
Inc : hash_func
const user_rserver, rserver_aserver,
aserver_user, sec1, sec2, sec3, sec4, sec5,
sec6 : protocol_id
init State :=0
transition
1. State = 0 ∧ Rcv(start) =|>
State' := 1 ∧ IDj' :=new()
∧ Snd({IDj'}_SKey1)
∧ Rcv({Ej.Dj}_SKey1)
2. State = 1 ∧ Rcv(M5') =|>
State' := 2 ∧ M9' := exp(M5',Dj)
∧ M9' := (IDi.Ri.M4)
∧ Yj' := h(Ej.Dj)
∧ Rj' := new()
∧ Rij' := xor(Ri,Rj)
∧ SKj' := h(IDi.IDj.Ri.Rj')
∧ M6' := h(SKj'.Rj'.IDj)
∧ Snd(M6'.Rij')
∧ secret({Ri,Rj'}, sec6, {ASj,Ui})
∧ witness(Ui, ASj, aserver_user, Rj)
end role
```

Algorithm 3: Role specification for the application-server of our protocol in HLPSL.

6. Performance Evaluation and Discussion

This section discusses the performance of the proposed protocol and then compares it with the recently published protocols [14, 16, 32, 33] in terms of computation, smart-card storage, and communication costs. Note that the login and authentication phases are necessary for performing secure communication in each authentication cycle. Therefore, we have considered these phases in comparison. To evaluate the computational cost, this article used the following time complexities.

(i) T_h: cost for one-way hash operation

(ii) T_{sym}: cost for symmetric key encryption/decryption operation

(iii) T_e: cost for exponentiation operation

(iv) T_m: cost for modular multiplication operation

The results obtained in [40] have executed various cryptographic operations using MIRACL C/C++ Library. It requires Windows 7 with 32-bit OS, Visual C++ 2008, 1024-bit cyclic group, 160-bit prime field F_p, AES for symmetric

```
role session(Ui, RC, ASj: agent,
SKey1 : symmetric_key,
SKey2 : symmetric_key,
H, MUL,SUB: hash_func)
def=
local SI, SJ, RI, RJ, PI, PJ: channel (dy)
composition
user(Ui, RC, ASj, SKey1, SKey2, H, MUL, SUB, SI, RI)
∧ rserver(Ui, RC, ASj, SKey1, SKey2, H, MUL, SUB, SJ, RJ)
∧ aserver(Ui, RC, ASj, SKey1, SKey2, H, MUL, SUB, PI, PJ)
end role
role environment()
def=
const ui, rc, asj: agent,
skey1 : symmetric_key,
skey2 : symmetric_key,
h,mul, sub: hash_func,
idi, pwi, idj, ai, bi, ci, di, ei, fi, m1, m2, m3, m4, m5, m6, ri, rj,
rij, ski, skj, n : text,
user_aserver_Ri, aserver_user_Rj, sec1, sec2, sec3, sec4, sec5,
sec6: protocol_id
intruder_knowledge = {ui, rc, asj, h, mul, m6, rij, m5, m3, di,
ei, fi, n, m1, m2, idj}
composition
session(rc, ui, asj, skey1, skey2, h, mul, sub)
∧ session(ui, rc, asj, skey1, skey2, h, mul, sub)
∧ session(asj, ui, rc, skey1, skey2, h, mul, sub)
end role
goal
secrecy_of sec1
secrecy_of sec2
secrecy_of sec3
secrecy_of sec4
secrecy_of sec5
secrecy_of sec6
authentication_on user_aserver_Ri
authentication_on aserver_user_Rj
end goal
environment()
```

Algorithm 4: Role specification for the goal, session, and environment of our protocol in HLPSL.

encryption/decryption, and SHA-1 operation. On these above configurations, execution time (in ms) of such different operations is as follows: $T_h \approx 0.0004$ ms, $T_{\text{sym}} \approx 0.1303$ ms, $T_m \approx 0.0147$ ms, and $T_e \approx 1.8269$ ms.

In Tables 2 and 3, we observed that computational cost and execution time of our protocol are lower than the protocols proposed in [14, 16] and nearly equal to the protocols in [32, 33]. Our protocol provides strong security protection against known attacks, such as mutual authentication, user anonymity, session key agreement, efficient login phase, and verification phase, and response time of the protocol is comparatively lesser than other protocols. For better understanding, we give execution cost (ms) comparison of the proposed protocol and the related ones [14, 16, 32, 33] in Figure 7.

In Table 4, smart-card storage cost, communication cost, and total number of message transmissions have been provided. The notation $160 \times S_j^*$ indicates that smart-card storage capacity is dependent on the total number of application servers. Therefore, the limitation of the protocols [14, 16, 32] is that the number of application servers is restricted owing to low capacity of smart-card. Note that smart-card storage cost does not increase with the number of application servers in the protocol [33] and our protocol. The proposed protocol requires less storage costs for the smart-card compared to the protocols proposed in [14, 16, 32, 33]. In our protocol, we provide session key verification property and to achieve it, the protocol takes extra single message exchange in comparison with the protocols [14, 16, 32, 33]. Note that the protocols [14, 16, 32, 33] do not provide the session key verification

Table 2: Computation cost comparison of our protocol with other protocols.

Scheme	User	Server	Total
Pippal et al. [14]	$4T_h + 3T_e + T_m$	$3T_h + 4T_e + T_m$	$7T_h + 7T_e + 2T_m$
Yeh [16]	$4T_h + 2T_e + T_m$	$5T_h + 4T_e + T_m$	$9T_h + 6T_e + 2T_m$
Wei et al. [32]	$7T_h + 2T_e$	$6T_h + 2T_e$	$13T_h + 4T_e$
Li et al. [33]	$5T_h + T_e$	$8T_h + 3T_e$	$13T_h + 4T_e$
Proposed	$6T_h + 2T_e$	$6T_h + 2T_e + T_{\text{sym}}$	$12T_h + 4T_e + T_{\text{sym}}$

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa/web-interface-computation/./
tempdir/workfile8Y6LVt.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00 s
  searchTime: 0.19 s
  visitedNodes: 17 nodes
  depth: 2 plies
```

Algorithm 5: OFMC back-end simulation result of our protocol.

Table 3: Execution time (ms) comparison of our protocol with other protocols.

Scheme	User	Server	Total
Pippal et al. [14]	5.4966 ms	7.3235 ms	12.8200 ms
Yeh [16]	3.6701 ms	7.5621 ms	11.2322 ms
Wei et al. [32]	3.6566 ms	3.6562 ms	7.3128 ms
Li et al. [33]	1.8289 ms	5.4839 ms	7.3128 ms
Proposed	3.6562 ms	3.7865 ms	7.4427 ms

property. In Table 4, we also noticed that the total number of bits transmission of our protocol is lesser than related protocols.

7. Conclusion and Future Scope

This article presents a flexible and cost-effective RSA-based multiserver authentication protocol to perform secure mutual authentication over any insecure channel with user and application server. We have then proved informally that our protocol can defy different cryptographic attacks. Moreover, the security of the mutual authentication and the freshness of the session key have been established based on BAN logic model, and AVISPA simulation results claim that proposed protocol is SAFE in CL-AtSe and OFMC

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/avispa/web-interface-computation
/./tempdir/workfilegQjd7wqdOJ.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed: 15 states
  Reachable: 15 states
  Translation: 0.32 seconds
  Computation: 0.00 seconds
```

Algorithm 6: CL-AtSe back-end result of the proposed protocol.

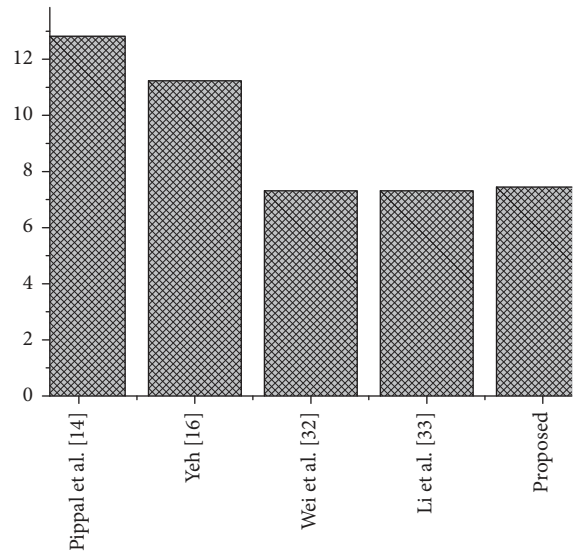


Figure 7: Execution time (ms) of different protocols.

models. We have then shown that the performance of our protocol is more effective than the existing protocols in terms of complexities. Our protocol is not only robust, but also flexible since it has user-friendly password change phase.

In recent times, cloud computing is also applied to the environment of mobile communication. However, the

Table 4: Communication and storage costs of our protocol and others.

Scheme	Smart-card cost (bits)	Communication cost (bits)	Number of messages
Pippal et al. [14]	$4256 + (160 \times S_j)^*$	2528	3
Yeh [16]	$4256 + (160 \times S_j)^*$	2528	3
Wei et al. [32]	$2368 + (160 \times S_j)^*$	3328	3
Li et al. [33]	3712	2688	3
Proposed	2528	2304	4

computation and storage capabilities of mobile devices are very limited. Therefore an attempt can be made to design a provably secure and lightweight multiserver authentication protocol for mobile cloud computing environments.

Notations

(i) List of Used Symbols

U_i: ith user
RC: Registration center
AS_j: jth application server
PW_i: Password of U_i
ID_i: Identity of U_i
ID_j: Identity of AS_j
R_i: Random number selected in login phase by the U_i
R_j: Random number selected in authentication phase by the AS_j
x: Secret key of RC
p, q: Two different large prime numbers
e_j: Public key of AS_j
d_j: Private key of AS_j
SK_i: Session key shared between U_i and AS_j
$h(\cdot)$: Cryptographic hash function
$\|$: Concatenation
$\oplus$: Bit-wise XOR operation
$ENC_x(M)$: Symmetric key encryption using key x of message M.

(ii) BAN Statements Used in This Paper

$P \mid\equiv X$: P believes X, or P would be entitled to believe X. In particular, P can take X as true
$P \triangleleft X$: P sees X. P has received some message X and is capable of reading and repeating it (seeing rule)
$P \mid\sim X$: P once said X. P at some time sent a message including the statement X. It is not known whether this is a replay, though it is known that P believed X when he sent it
$P \Rightarrow X$: P has jurisdiction over X. The principal P is an authority on X and should be trusted on this matter
$\sharp(X)$: Message X is fresh
(X, Y): Formula X or Y is one part of the formulae (X, Y)
$\langle X \rangle_Y$: Formula X is combined with the formulae Y
$\{X\}_K$: Formula X is encrypted under the key K
$(X)_K$: Formula X is hashed with the key K
$P \overset{K}{\longleftrightarrow} Q$: Principals P and Q communicate via shared key K
$P \overset{X}{\rightleftharpoons} Q$: Formula X is a secret only known to P and $Q$and possibly to principals trusted by them
$\overset{K}{\mapsto} P$: Principal P has K as its public key
SK: Session key is used in the current session.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through Research Group no. RG-288.

References

[1] R. Amin and G. P. Biswas, "Anonymity preserving secure hash function based authentication scheme for consumer USB mass storage device," *In Proceedings of the Computer, Communication, Control and Information Technology (C3IT'15)*, pp. 1–6, 2015.

[2] R. Amin, "Cryptanalysis and an efficient secure ID-based remote user authentication scheme using smart card," *International Journal of Computer Applications*, vol. 75, no. 13, pp. 43–48, 2013.

[3] R. Amin, T. Maitra, and S. Rana, "An improvement of Wang. et. al.'s remote user authentication scheme against smart card security breach," *International Journal of Computer Applications*, vol. 75, no. 13, pp. 37–42, 2013.

[4] R. Amin and G. P. Biswas, "Cryptanalysis and design of a three-party authenticated key exchange protocol using smart card," *Arabian Journal for Science and Engineering*, vol. 40, pp. 31–35, 2015.

[5] S. H. Islam, G. P. Biswas, and K.-K. R. Choo, "Cryptanalysis of an Improved Smartcard-based Remote Password Authentication Scheme," *Information Sciences Letter*, vol. 3, no. 1, pp. 35–40, 2014.

[6] R. Amin, S. H. Islam, G. P. Biswas, and M. K. Khan, "An efficient remote mutual authentication scheme using smart mobile phone over insecure networks," in *Proceedings of the International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA'15)*, pp. 1–7, 2015.

[7] R. Amin and G. P. Biswas, "An improved RSA based user authentication and session key agreement protocol usable in TMIS," *Journal of Medical Systems*, vol. 39, no. 8, pp. 1–14, 2015.

[8] D. Giri, T. Maitra, R. Amin, and P. D. Srivastava, "An efficient and robust rsa-based remote user authentication for telecare medical information systems," *Journal of Medical Systems*, vol. 39, no. 145, 2015.

[9] R. Amin and G. P. Biswas, "Remote Access Control Mechanism Using Rabin Public Key Cryptosystem," in *Proceedings of the In Proceedings of the Information Systems Design and Intelligent Applications*, pp. 525–533, 2015.

[10] R. Amin and G. P. Biswas, "A secure three-factor user authentication and key agreement protocol for TMIS with user anonymity," *Journal of Medical Systems*, vol. 39, no. 8, pp. 1–19, 2015.

[11] S. H. Islam, M. K. Khan, M. S. Obaidat, and F. T. B. Muhaya, "Provably secure and anonymous password authentication protocol for roaming service in global mobility networks using extended chaotic maps," *Wireless Personal Communications*, vol. 84, no. 3, pp. 2013–2034, 2015.

[12] S. H. Islam, "Design and analysis of a three party password-based authenticated key exchange protocol using extended chaotic maps," *Information Sciences. An International Journal*, vol. 312, pp. 104–130, 2015.

[13] R. Amin and G. P. Biswas, "Design and Analysis of Bilinear Pairing Based Mutual Authentication and Key Agreement Protocol Usable in Multi-server Environment," *Wireless Personal Communications*, vol. 84, no. 1, pp. 439–462, 2015.

[14] R. S. Pippal, C. Jaidhar, and S. Tapaswi, "Robust smart card authentication scheme for multiserver architecture," *Wireless Personal Communications*, vol. 72, no. 1, pp. 729–745, 2013.

[15] S. K. Sood, A. K. Sarje, and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 609–618, 2011.

[16] K.-H. Yeh, "A Provably Secure Multi-server Based Authentication Scheme," *Wireless Personal Communications*, vol. 79, no. 3, pp. 1621–1634, 2014.

[17] X. Li, Y. Xiong, J. Ma, and W. Wang, "An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 763–769, 2012.

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

[19] S. H. Islam, "A provably secure ID-based mutual authentication and key agreement scheme for mobile multi-server environment without ESL attack," *Wireless Personal Communications*, 2014.

[20] R. Amin and G. P. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Networks*, vol. 36, no. 1, pp. 58–80, 2016.

[21] R. Amin and G. P. Biswas, "A Novel User Authentication and Key Agreement Protocol for Accessing Multi-Medical Server Usable in TMIS," *Journal of Medical Systems*, vol. 39, no. 3, pp. 1–17, 2015.

[22] Y-P. Liao and S-S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards and Interfaces*, vol. 31, no. 1, pp. 24–29, 2009.

[23] T-Y. Chen, M-S. Hwang, C-C. Lee, and J-K. Jan, "Cryptanalysis of a secure dynamic id based remote user authentication scheme for multi-server environment," in *Proceedings of the In Proceedings of the fourth international conference on innovative computing information and control (ICICIC*, pp. 725–728, 2009.

[24] H-C. Hsiang and W-K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards and Interfaces*, vol. 31, no. 6, pp. 1118–1123, 2009.

[25] C-C. Lee, T-H. Lin, and R-X. Chang, "A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13863–13870, 2011.

[26] T-T. Truong, M-T. Tran, and A-D. Duong, "Robust secure dynamic id based remote user authentication scheme for multi-server environment," in *Proceedings of the In Proceedings of the Computational science and its applications (ICCSA13*, pp. 502–515, 2013.

[27] D. He, J. Chen, W. Shi, and M. K. Khan, "On the security of an authentication scheme for multiserver architecture," *International Journal of Electronic Security and Digital Forensics*, vol. 5, no. 3, pp. 288–296, 2013.

[28] W. B. Hsieh and J. S. Leu, "An anonymous mobile user authentication protocol using self-certified public keys based on multi-server architectures," *Journal of Supercomputing*, vol. 70, no. 1, pp. 133–148, 2014.

[29] Y. P. Liao and C. M. Hsiao, "A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients," *Future Generation Computer Systems*, vol. 29, no. 3, pp. 886–900, 2013.

[30] A. K. Sutrala, A. K. Das, V. Odelu, M. Wazid, and S. Kumari, "Secure anonymity-preserving password-based user authentication and session key agreement scheme for telecare medicine information systems," *Computer Methods and Programs in Biomedicine*, vol. 135, pp. 167–185, 2016.

[31] H. Arshad and A. Rasoolzadegan, "Design of a Secure Authentication and Key Agreement Scheme Preserving User Privacy Usable in Telecare Medicine Information Systems," *Journal of medical systems*, vol. 40, no. 11, p. 237, 2016.

[32] J. Wei, W. Liu, and X. Hu, "Cryptanalysis and Improvement of a Robust Smart Card Authentication Scheme for Multi-server Architecture," *Wireless Personal Communications*, vol. 77, no. 3, pp. 2255–2269, 2014.

[33] X. Li, J. Niu, S. Kumari, J. Liao, and W. Liang, "An enhancement of a smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 80, no. 1, pp. 175–192, 2015.

[34] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO'99)*, vol. 1666, pp. 388–397, 1999.

[35] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–555, 2002.

[36] Y-F. Chang, S-H. Yu, and D-R. Shiao, "An uniqueness-and anonymity- preserving remote user authentication scheme for connected health care," *Journal of Medical Systems*, vol. 37, no. 9902, 2013.

[37] A. Armando, D. Basin, Y. Boichut et al., "The AVISPA tool for the automated validation of internet security protocols and applications," in *Proceedings of the 17th International conference on computer aided verification (CAV05), LNCS*, vol. 3576, pp. 281–285, 2005.

[38] AVISPA., "Automated validation of internet security protocols and applications," http://www.avispa-project.org/, 2014.

[39] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 2009.

[40] L. Xu and F. Wu, "Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care," *Journal of Medical Systems*, vol. 39, no. 10, 2015.