IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 17, 2019, accepted July 12, 2019, date of publication July 31, 2019, date of current version August 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2932181

A Robust Authentication Scheme for Multiple Servers Architecture

HUAWEI WANG[1], DIANLI GUO[2], QIAOYAN WEN[1], AND HUA ZHANG[1]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]The 6th Research Institute of China Electronics Corporation, Beijing 100083, China

Corresponding authors: Dianli Guo (guodianli@163.com), Qiaoyan Wen (wqy@bupt.edu.cn), and Hua Zhang (zhanghua_288@bupt.edu.cn)

ABSTRACT Multiple servers collaboration technology furnishes an effective storage management platform to the service providers. The massive data, which belong to disparate providers, possesses distinct access policies for each user. Authentication applied to multiple servers architecture is recognized as a remarkable mechanism for access control and authorization of users. In 2017, Jangirala *et al.* explored problems existing in the current research of authentication for multiple servers environment and presented a new solution. Nevertheless, we find that it is defenseless in impersonation attack, server spoofing attack, and fails to maintain users' anonymity. In this paper, we detail the aforementioned faults and propose a remedy with a tripartite certification strategy. Correspondingly, the validation of BAN-logic focused on tripartite authentication protocol is put forward to demonstrate the security reliability. Comparative evaluation of other related solutions for the multiple servers architecture observes that our proposal has advantages over security attributes meanwhile provides a reliable guarantee of efficiency.

INDEX TERMS Anonymity, authentication, BAN-logic, multiple servers architecture.

I. INTRODUCTION

The emergence and vigorous development of the cloud computing unprecedentedly change the pattern of the traditional network information services. An increasing number of enterprises have encapsulated the data and services into the clouds, instead of sustaining the infrastructures of local database servers. It significantly facilitates the development of enterprises, which concentrate on their core services with cloud solutions. The cloud computing is the next paradigm for computing and profoundly effects everyone, however at the meantime it also brings about a series of privacy and security concerns. Each service provider stores the massive amounts of data in the cloud, which is draw up a unique access policy for users [1], [2]. Analogously, each user may obtain network resources belonging to multiple service providers on the cloud computing platform, the tautological registration problems from which have been unfolded incrementally. Thereby, it has become an irresistible trend to formulate a unified authentication frame for such a. Multiple servers authentication mechanism enables users to access network services located on distributed servers with single registration and it is a feasible solution to solve the aforementioned issue [7].

In the initialization of the multiple servers authentication system, the registration center is in charge of producing private keys for servers and users with the created primary secret key. After that, servers and users get their respective private keys to complete the authentication and key agreement protocol. Servers expect that the network services are accessed by authorized users, and the legitimate users want to enjoy convenient services with privacy protections [9], [10]. Note that, the centralized registration tactic enables that servers and users register once, and then users can log in each server with same userid/password [8].

The authentication mode in such a multiple servers system can be basically divided into two different types: two-party authentication mode [13], [14], [16]–[21] and three-party authentication mode [10]–[12], [15], [22]–[25]. As the name suggests, the former is only implemented by users and servers. The latter one permits the registration center to take part in each round of authentication to verify the validity of the other two participants. Concretely, users have to store all the secret keys associated with each server in the portable device for realizing mutual authentication in the two-party

The associate editor coordinating the review of this manuscript and approving it for publication was Nafees Mansoor.

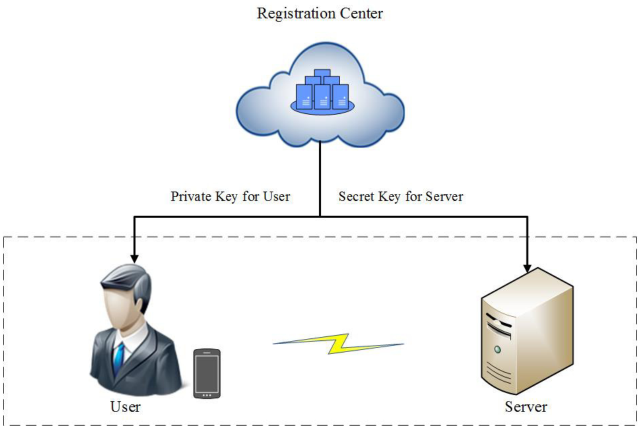

FIGURE 1. Two-party authentication mode.

authentication mode. The storage capacity for private keys of clients is added linearly as the number of servers increases. The three-party authentication mode sacrifices the computing efficiency in exchange for better storage capacity of users. This mode increases calculation quantity of the registration center which owns an enormous computing resources, rather than the users' portable devices with limited processing power. Obviously, the latter mode would be more applicable to wireless mobile environment. However, it's important to point out that the registration center is incapable of acquiring the shared session key between users and servers in either authentication mode.

In the recent paper, Jangirala *et al.* [28] analyzed Shunmuganathan *et al.*'s [26] multi-server authentication protocol and proposed a new scheme with the two-party authentication mode. Although Jangirala et al.'s scheme enhanced the efficiency of the previously proposed schemes, we find that their scheme is exposed to impersonation attack, server spoofing attack and privacy disclosure. Therefore, we devise a modified authentication scheme for multiple servers architecture with three-party authentication mode to address the identified shortcomings. We further present the formal security analysis based on BAN-logic to demonstrate our scheme achieving complete triple authentication. The performance evaluation shows that the improved scheme is superior than the predecessor protocol in the security properties, meanwhile maintains an acceptable execution efficiency.

The rest of the paper is organized as follows. The review and security analysis of Jangirala et al.'s scheme are presented in Section 2 and 3, respectively. In Section 4, we detail the proposed solution for multiple servers architecture. The rigorous validation based on BAN-logic and the informal security analysis are provided in Section 5. Subsequently, the performance comparison with related protocols is illustrated in Section 6. Finally, we conclude this paper in Section 7.

Herein, we define notations in Table 1 used in the whole article.

TABLE 1. Notations.

U_i	denotes user
S_j	denotes the service providing server
RC	denotes the registration center
ID_i	denotes U_i's username
PW_i	denotes U_i's password
SID_j	denotes S_j's identity
x	denotes the primary secret key of the system
SK	denotes the shared session key
$H(\cdot)$	denotes collision resistant hash function
$E_k(M)$	denotes encrypting a message M with k
$D_k(C)$	denotes decrypting a ciphertext C with k

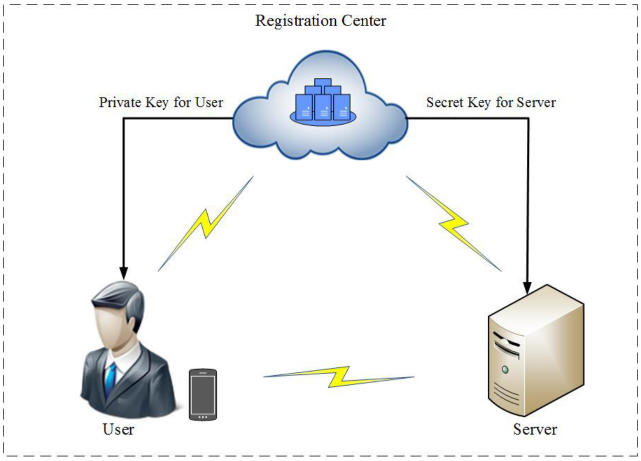

FIGURE 2. Three-party authentication mode

II. REVIEW OF JANGIRALA ET AL.'S SCHEME

Four main phases in Jangirala et al.'s scheme are registration phase, login phase, authentication and key agreement phase, password change phase. In Fig 3, we further illustrate their scheme except the preprocessed registration phase and separate executed password change phase.

Initially, the registration center RC chooses its primary secret key x and random number y to compute $H(x\|y)$ and $H(y)$. After that, shares them with service providing servers S_j via a secure communication channel. Both x and y should be hold for safekeeping by RC.

A. REGISTRATION PHASE

Step 1: U_i chooses a random number b and creates $A_i = H(ID_i \oplus b \oplus PW_i)$ with the username ID_i and the corresponding password PW_i. Then, he/she register in the registration center RC with $\{ID_i, A_i\}$, which includes U_i's hidden credential.

Step 2: RC receives U_i's registration request securely and computes $B_i = H(A_i\| x)$, $C_i = H(ID_i\|H(y)\|A_i)$, $D_i = H(B_i\|H(x\|y))$, $E_i = B_i \oplus H(x\|y)$. And then, RC issues a smart card for U_i, which is stored with the computed $\{C_i, D_i, E_i, H(\cdot), H(y)\}$.

Step 3: U_i receives the issued smart card and keys $L_i = b \oplus H(ID_i\|PW_i)$ into its memory.

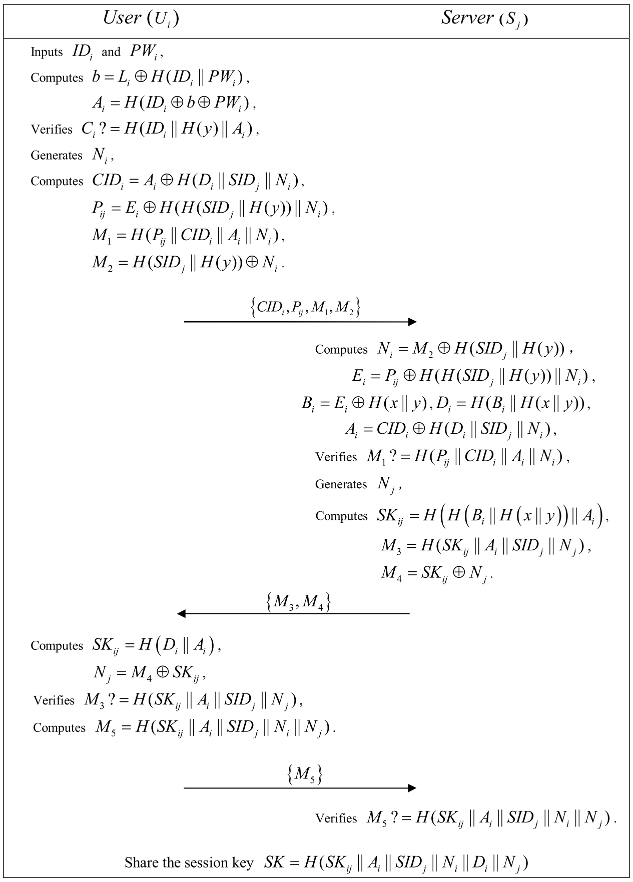
User (U_i)
Server (S_j)
Inputs ID_i and PW_i,
Computes $b = L_i \oplus H(ID_i \| PW_i)$,
$A_i = H(ID_i \oplus b \oplus PW_i)$,
Verifies $C_i ?= H(ID_i \| H(y) \| A_i)$,
Generates N_i,
Computes $CID_i = A_i \oplus H(D_i \| SID_j \| N_i)$,
$P_{ij} = E_i \oplus H(H(SID_j \| H(y)) \| N_i)$,
$M_1 = H(P_{ij} \| CID_i \| A_i \| N_i)$,
$M_2 = H(SID_j \| H(y)) \oplus N_i$.
$\{CID_i, P_{ij}, M_1, M_2\}$
Computes $N_i = M_2 \oplus H(SID_j \| H(y))$,
$E_i = P_{ij} \oplus H(H(SID_j \| H(y)) \| N_i)$,
$B_i = E_i \oplus H(x \| y), D_i = H(B_i \| H(x \| y))$,
$A_i = CID_i \oplus H(D_i \| SID_j \| N_i)$,
Verifies $M_1 ?= H(P_{ij} \| CID_i \| A_i \| N_i)$,
Generates N_j,
Computes $SK_{ij} = H\left(H\left(B_i \| H\left(x \| y\right)\right) \| A_i\right)$,
$M_3 = H(SK_{ij} \| A_i \| SID_j \| N_j)$,
$M_4 = SK_{ij} \oplus N_j$.
$\{M_3, M_4\}$
Computes $SK_{ij} = H\left(D_i \| A_i\right)$,
$N_j = M_4 \oplus SK_{ij}$,
Verifies $M_3 ?= H(SK_{ij} \| A_i \| SID_j \| N_j)$,
Computes $M_5 = H(SK_{ij} \| A_i \| SID_j \| N_i \| N_j)$.
$\{M_5\}$
Verifies $M_5 ?= H(SK_{ij} \| A_i \| SID_j \| N_i \| N_j)$.
Share the session key $SK = H(SK_{ij} \| A_i \| SID_j \| N_i \| D_i \| N_j)$

FIGURE 3. Login phase and authentication & key agreement phase in Jangirala *et al.*'s scheme.

B. LOGIN PHASE

Step 1: U_i places the smart card into the terminal and is requested to input his/her legitimate username ID_i and password PW_i.

Step 2: The smart card employs ID_i and PW_i to calculate $b = L_i \oplus H(ID_i \| PW_i)$, $A_i = H(ID_i \oplus b \oplus PW_i)$, $C_i^* = H(ID_i \| H(y) \| A_i)$. Then, it checks whether C_i^* equals to the stored C_i to verify the validity of the inputted username and password. If $C_i \neq C_i^*$, the smart card terminates the login phase and indicates that the username and password do not match.

Step 3: U_i keys in the identity SID_j of the target server S_j, and then the smart card generates a random number N_i to calculate $CID_i = A_i \oplus H(D_i \| SID_j \| N_i)$, $P_{ij} = E_i \oplus H(H(SID_j \| H(y)) \| N_i)$, $M_1 = H(P_{ij} \| CID_i \| A_i \| N_i)$, $M_2 = H(SID_j \| H(y)) \oplus N_i$.

Step 4: Finally, the smart card submits the login request $\{CID_i, P_{ij}, M_1, M_2\}$ to the target server S_j.

C. AUTHENTICATION AND KEY AGREEMENT PHASE

Step 1: S_j obtains the login request $\{CID_i, P_{ij}, M_1, M_2\}$ from U_i, S_j computes $N_i = M_2 \oplus H(SID_j \| H(y))$, $E_i = P_{ij} \oplus H(H(SID_j \| H(y)) \| N_i)$, $B_i = E_i \oplus H(x \| y)$, $D_i = H(B_i \| H(x \| y))$ and $A_i = CID_i \oplus H(D_i \| SID_j \| N_i)$ with the known credential $H(y)$ and $H(x \| y)$.

Step 2: In order to validate whether U_i has authorization to access, S_j calculates $M_1^* = H(P_{ij} \| CID_i \| A_i \| N_i)$ and verifies $M_1^* ? = M_1$. If the equation does not hold, S_j rejects U_i's login request; on the contrary, S_j goes on producing a random number N_j to compute $SK_{ij} = H(H(B_i \| H(x \| y)) \| A_i)$, $M_3 = H(SK_{ij} \| A_i \| SID_j \| N_j)$, $M_4 = SK_{ij} \oplus N_j$. Subsequently, the replied message $\{M_3, M_4\}$ is sent to U_i for approving the permission.

Step 3: U_i needs to confirm that the permission messages $\{M_3, M_4\}$ from S_j is accurate. Then U_i computes $SK_{ij} = H(D_i \| A_i)$, $N_j = M_4 \oplus SK_{ij}$, $M_3^* = H(SK_{ij} \| A_i \| SID_j \| N_j)$ and compares M_3^* with the received M_3. If $M_3^* \neq M_3$, it means that the received messages is inaccurate; else, U_i confirms the access authorization. Afterwards, U_i computes $M_5 = H(SK_{ij} \| A_i \| SID_j \| N_i \| N_j)$ and sends it to S_j for session key negotiation.

Step 4: S_j computes $H(SK_{ij} \| A_i \| SID_j \| N_i \| N_j)$ to verify that it is consistent with the received M_5. If so, S_j notarize U_i is authorized accessing user.

After the above authentication process, U_i and S_j negotiate a session key $SK = H(SK_{ij} \| A_i \| SID_j \| N_i \| D_i \| N_j)$.

D. PASSWORD CHANGE PHASE

Step 1: If U_i wish to change password, the smart card force him/her to key ID_i, PW_i by means of a security policy.

Step 2: The smart card computes $b = L_i \oplus H(ID_i \| PW_i)$, $A_i = H(ID_i \oplus b \oplus PW_i)$, $C_i^* = H(ID_i \| H(y) \| A_i)$, and checks whether C_i^* equals to C_i. If $C_i^* = C_i$, the smart card proceeds to execute this phase and enables users to enter a new password PW_i^{new}.

Step 3: After that, the smart card computes $A_i^{new} = H(ID_i \oplus b \oplus PW_i^{new})$, $C_i^{new} = H(ID_i \| H(y) \| A_i^{new})$, $L_i^{new} = b \oplus H(ID_i \| PW_i^{new})$. Finally, it replaces C_i and L_i with recalculated one and realizes the password update.

III. CRYPTANALYSIS OF JANGIRALA ET AL.'S SCHEME

In this section, we demonstrate that Jangirala *et al.*'s scheme [28] is susceptible to impersonation attack, server spoofing attack and user privacy disclosure. As mentioned before, it is essential that user stores all the secret keys associated with each server in the the two-party authentication mode. In their scheme, user applies for accessing the different servers with the invariable secret value. This is the main ingredient which results all the vulnerabilities. The details of these flaws are described as follows.

A. USER PRIVACY DISCLOSURE

Any adversary $\mathcal{A}$ could extract the value $H(y)$ from the smart card and eavesdrop the login request message $\{CID_i, P_{ij}, M_1, M_2\}$ of legitimate user U_i. Then $\mathcal{A}$ employs the above values to compute $N_i = M_2 \oplus H(SID_j \| H(y))$, $E_i = P_{ij} \oplus H(H(SID_j \| H(y)) \| N_i)$.

The value E_i is structurally constant and is a specific number associated with U_i's username ID_i. It is deemed to be a recognition number for U_i to login servers. $\mathcal{A}$ could track the history access records of the victim user and steal his/her privacy information. As a result, the above attack is effective and practical.

B. IMPERSONATION ATTACKS

Resistance to impersonation attack is the fundamental security property for authentication schemes. If any adversary $\mathcal{A}$ could impersonate other legal users to access servers, it doesn't make sense to implement access control and authorization. In the following, we examine this attack in detail.

Firstly, the adversary extracts values $H(y)$ and D_i from U_i's smart card by side-channel attack [3], [4]. Afterwards, $\mathcal{A}$ transmits the previously U_i's login request $\{CID_i, P_{ij}, M_1, M_2\}$, which is intercepted in the public communication channel to S_j. It is worth noting that the previous login request is valid although it is overdue. Once the login request is received, the target server S_j will reply message $\{M_3, M_4\}$ for approving the permission. Secondly, the adversary computes $N_i = M_2 \oplus H(SID_j \| H(y))$ and $A_i = CID_i \oplus H(D_i \| SID_j \| N_i)$, $M_5 = H(SK_{ij} \| A_i \| SID_j \| N_i \| N_j) = H(H(D_i \| A_i) \| A_i \| SID_j \| N_i \| N_j)$. Then the adversary transmits M_5 to S_j. It's easy to see that the adversary successfully impersonates as U_i to deceive the service providing server through the above operation.

C. SERVER SPOOFING ATTACK

Phishing is a common server spoofing attack, which compromises the accounts of online banking customers as well as other personal information. Therefore, defence such an attack is fateful for the security of authentication protocols. In the following, we prove that Jangirala et al.'s scheme suffers from server spoofing attack.

Let S_k be a malicious server which possesses secret keys $H(y)$ and $H(x \| y)$. These values are identical for all servers administered by RC. After intercepting the login request message $\{CID_i, P_{ij}, M_1, M_2\}$ of U_i, S_k can calculate $N_i = M_2 \oplus H(SID_j \| H(y))$, $E_i = P_{ij} \oplus H(H(SID_j \| H(y)) \| N_i)$, $B_i = E_i \oplus H(x \| y)$, $D_i = H(B_i \| H(x \| y))$, $A_i = CID_i \oplus H(D_i \| SID_j \| N_i)$, $SK_{ij} = H(H(B_i \| H(x \| y)) \| A_i)$, $M_3' = H(SK_{ij} \| A_i \| SID_j \| N_j')$, $M_4 = SK_{ij} \oplus N_j'$, where N_j' is a randomly selected number. Afterwards, sends the forged reply message $\{M_3', M_4'\}$ to U_i. Obviously, the response message is correct and could pass the validation.

IV. OUR SCHEME

In this section, we present an enhanced protocol for multiple servers architecture with a three-party authentication mode, which can remedy the identified security flaws. In the proposed protocol, the private keys size of users and servers are both $O(1)$. The registration center undertakes a portion of verification tasks of service providing servers and users for trade off between efficiency and security. Our proposal includes five phases which are consistent with the original scheme except the server registration phase. In Fig 4, we depict the main phases including login phase and authentication & key agreement phase.

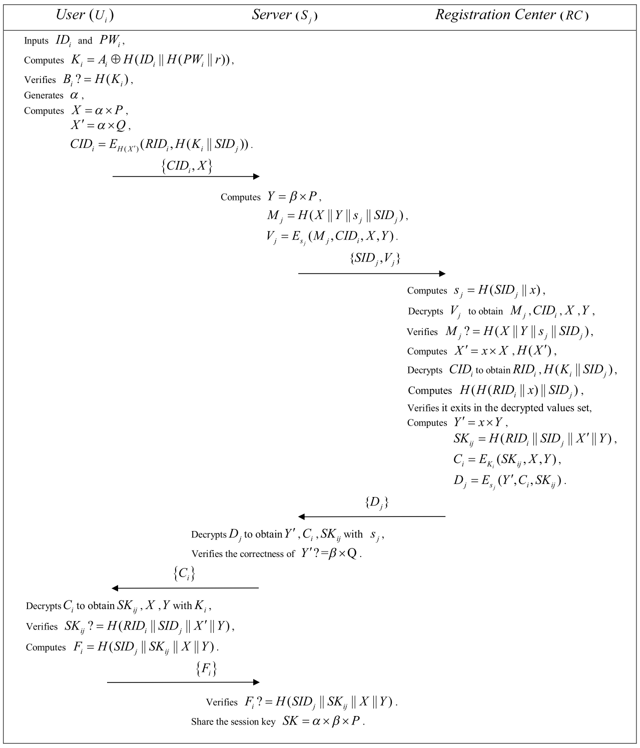
User (U_i)
Server (S_j)
Registration Center (RC)
Inputs ID_i and PW_i,
Computes $K_i = A_i \oplus H(ID_i \| H(PW_i \| r))$,
Verifies $B_i ?= H(K_i)$,
Generates α,
Computes $X = \alpha \times P$,
$X' = \alpha \times Q$,
$CID_i = E_{H(X')}(RID_i, H(K_i \| SID_j))$.
$\{CID_i, X\}$
Computes $Y = \beta \times P$,
$M_j = H(X \| Y \| s_j \| SID_j)$,
$V_j = E_{s_j}(M_j, CID_i, X, Y)$.
$\{SID_j, V_j\}$
Computes $s_j = H(SID_j \| x)$,
Decrypts V_j to obtain M_j, CID_i, X, Y,
Verifies $M_j ?= H(X \| Y \| s_j \| SID_j)$,
Computes $X' = x \times X, H(X')$,
Decrypts CID_i to obtain $RID_i, H(K_i \| SID_j)$,
Computes $H(H(RID_i \| x) \| SID_j)$,
Verifies it exits in the decrypted values set,
Computes $Y' = x \times Y$,
$SK_{ij} = H(RID_i \| SID_j \| X' \| Y)$,
$C_i = E_{K_i}(SK_{ij}, X, Y)$,
$D_j = E_{s_j}(Y', C_i, SK_{ij})$.
$\{D_j\}$
Decrypts D_j to obtain Y', C_i, SK_{ij} with s_j,
Verifies the correctness of $Y' ?= \beta \times Q$.
$\{C_i\}$
Decrypts C_i to obtain SK_{ij}, X, Y with K_i,
Verifies $SK_{ij} ?= H(RID_i \| SID_j \| X' \| Y)$,
Computes $F_i = H(SID_j \| SK_{ij} \| X \| Y)$.
$\{F_i\}$
Verifies $F_i ?= H(SID_j \| SK_{ij} \| X \| Y)$.
Share the session key $SK = \alpha \times \beta \times P$.

FIGURE 4. **Login phase and authentication & key agreement phase in our scheme.**

In the initialization of multiple servers authentication system, the registration center RC firstly chooses a large prime p and generates an elliptic curve group with generator P of order p [5]. Subsequently, RC continues to select a random number $x \in Z_p$ as the primary secret key, the public key is $Q = x \cdot P \bmod p$.

A. SERVER REGISTRATION PHASE

Our proposed multiple servers authentication system is extensible and flexible, that is, servers can apply for registration in RC for providing services at any time rather than merely in the initialization of the system. It is superior over the multiple servers system based on two-party authentication mode.

Step 1: S_j transmits the selected identity SID_j to RC in plaintext format via an authenticated communication channel.

Step 2: RC receives the identity SID_j and computes $s_j = H(SID_j \| x)$ for S_j. Then, RC sends the authorization value s_j to S_j.

Step 3: On receiving s_j, S_j officially becomes a legitimate service providing server.

B. USER REGISTRATION PHASE

The expandability of the proposed protocol is facilitated not just for servers registration but for the users registration as well. Users also register in RC at anytime by performing the following operations.

Step 1: U_i chooses number $r \in Z_p$ randomly and registers in RC with username ID_i, $RPW_i = H(PW_i\|r)$ where include the selected password PW_i implicitly. $\{ID_i, RPW_i\}$ is finally transmitted to RC over an authenticated communication channel.

Step 2: RC receives U_i's registration request and computes $RID_i = H(ID_i\|R_i)$, $K_i = H(RID_i\|x)$, $A_i = K_i \oplus H(ID_i\|RPW_i)$ and $B_i = H(K_i)$, where $R_i \in Z_p$ is a unique random number for each user. Subsequently, RC responses to U_i with the credentials $\{A_i, B_i, P, Q, H(\cdot), E_k\}$ securely.

Step 3: Upon receiving these secret values, U_i keys them with r into his/her portable device, such as mobile phones or PDAs. Obviously, the secret key size of user U_i is constant.

C. LOGIN PHASE

Step 1: U_i enters username ID_i and password PW_i into the portable device for confirming their correctness locally. Then the portable device computes $K_i = A_i \oplus H(ID_i\|H(PW_i\|r))$, $B_i^* = H(K_i)$ and verifies whether $B_i^* = B_i$ or not. If the equation holds, proceeds to Step 2; otherwise, it prompts that the typed username and password is incorrect and re-entered them. Noticeably, three consecutive wrong attempts will trigger locking mechanism about ten minutes.

Step 2: After the correctness validation of the entered ID_i and PW_i, the portable device proceeds to compute $X = \alpha \times P$, $X' = \alpha \times Q$, $CID_i = E_{H(X')}(RID_i, H(\ K_i\|SID_j))$, where $\alpha \in Z_p$ is a generated random number. Finally, U_i sends the login request message $\{CID_i, X\}$ to the target server S_j.

D. AUTHENTICATION AND KEY AGREEMENT PHASE

Step 1: S_j receives the login request and generates a random integer number $\beta \in Z_p$ to calculate $Y = \beta \times P$, $M_j = H(X\|Y\|s_j\|SID_j)$, $V_j = E_{s_j}(M_j, CID_i, X, Y)$. Afterwards, S_j transmits the message $\{SID_j, V_j\}$ to RC, which encapsulates U_i's request with its secret key.

Step 2: RC receives the message $\{SID_j, V_j\}$ and computes $s_j = H(SID_j\|x)$ to decapsulate M_j, CID_i, X, Y from V_j. Subsequently, RC is commissioned to ensure the legitimacy of S_j with verifying the consistence of the computed $H(X\|Y\|s_j\|SID_j)$ and the decrypted M_j. If the verification fails, RC denies executing the next step and reports an error.

Step 3: Next, RC proceeds to decrypt CID_i to recover RID_i, $H(K_i\|SID_j)$ with $H(x \times X)$. Then, RC computes $K_i = H(RID_i\|x)$, $H(K_i\|SID_j)$ and ensures that the computed $H(K_i\|SID_j)$ exits in the decrypted values set as expected. If the verification fails, RC aborts the request and returns a failure message; else, RC accomplishes the authentication for S_j and U_i successively.

Step 4: After that, RC computes $Y' = x \times Y$, $SK_{ij} = H(RID_i\|SID_j\|X'\|Y)$, $C_i = E_{K_i}(SK_{ij}, X, Y)$, $D_j = E_{s_j}(Y', C_i, SK_{ij})$ and sends the reply mutual authentication message $\{D_j\}$ to S_j.

Step 5: S_j receives the response message $\{D_j\}$ and recovers (Y', C_i, SK_{ij}) with secret key s_j. Subsequently, S_j checks the validity of the decrypted Y' by $Y'? = \beta \times Q$. If it is consistent with the calculated one, S_j assures that U_i is authentic by means of entrusted agency RC. Subsequently, S_j transmits another decrypted value $\{C_i\}$ to U_i.

Step 6: When receives $\{C_i\}$ from the target server S_j, U_i's portable device retrieves (SK_{ij}, X, Y) by decode key K_i. Afterwards, the portable device calculates $H(RID_i\|SID_j\|X'\|Y)$ and checks the equality of it and the decrypted SK_{ij}. If $SK_{ij} = H(RID_i\|SID_j\|X'\|Y)$, U_i confirms the access authorization and the legitimacy of S_j is ensured. Then, U_i computes $F_i = H(SID_j\|SK_{ij}\|X\|Y)$ and sends the response message $\{F_i\}$ to S_j for session key negotiation.

Step 7: S_j computes $F_i^* = H(SID_j\|SK_{ij}\|X\|Y)$ and checks whether $F_i^* = F_i$ or not after receiving the reply message. If they are equal. S_j notarizes that U_i is authorized accessing user and negotiates a session key $SK = H(SK_{ij}\|A_i\|SID_j\|N_i\| D_i\|N_j)$.

E. PASSWORD CHANGE PHASE

Step 1: U_i invokes the password update module and follows the prompts to key ID_i, PW_i for confirming validation locally.

Step 2: The portable device computes $K_i = A_i \oplus H(ID_i\|H(PW_i\|r))$ and checks $B_i? = H(K_i)$. If they are unequal, it prompts that the typed username and password do not match and re-imports them. As a security measure, the password update module will be locked if U_i enter the wrong username and password three times continuously.

Step 3: U_i is requested to input a new password twice. If the input passwords are consistent, The portable device computes $A_i^{new} = K_i \oplus H(ID_i\|H(PW_i^{new}\|r))$ and stores it into its memory to replace A_i. On the contrary, it reports typographical errors and asks U_i to re-enter a new password twice again.

V. SECURITY ANALYSIS AND DISCUSSION

A. AUTHENTICATION PROOF BASED ON BAN-LOGIC

In the previous section, we employed the three-party authentication mode to construct the multiple servers authentication protocol. Herein, we demonstrate that the proposed scheme is reasoned as secure by the formal proof method BAN-logic [6] for tripartite authentication protocol.

The notations and their implications are defined as follows.

1) $\mathcal{P} \mid\equiv X$: The entity $\mathcal{P}$ confirms that X is real and believable.
2) $\sharp(X)$: X be recognized as a fresh value which is never sent before.
3) $\mathcal{P} \Rightarrow X$: Each entity believes that the entity $\mathcal{P}$ has jurisdiction to generate the statement X.
4) $\mathcal{P} \lhd X$: The entity $\mathcal{P}$ receives message X and is permitted to handle it.
5) $\mathcal{P} \mid\sim X$: The entity $\mathcal{P}$ transmitted message X previously.
6) (X, Y): This message includes formulas X and Y.
7) $\langle X\rangle_Y$: This message is produced with the formula X combined with secret Y.

8) $\{X\}_Y$: This represents that X is encrypted with the secret key Y.
9) $\mathcal{P} \xleftrightarrow{K} \mathcal{Q}$: Entities $\mathcal{P}$ and $\mathcal{Q}$ shares a secret key K which is unrevealed for any untrustworthy one.
10) $\mathcal{P} \overset{K}{\rightleftharpoons} \mathcal{Q}$: Entities $\mathcal{P}$ and $\mathcal{Q}$ possess the identical secret K for affirming their credibility. K is also known to entities trusted by both of them.

Next, we present some logical postulates which are crucial for the validation of BAN-logic.

1) The message-meaning rule: $\frac{\mathcal{P}|\equiv\mathcal{Q}\xleftrightarrow{K}\mathcal{P},\mathcal{P}\lhd\{X\}_K}{\mathcal{P}|\equiv\mathcal{Q}|\sim X}$, $\frac{\mathcal{P}|\equiv\mathcal{Q}\overset{K}{\rightleftharpoons}\mathcal{P},\mathcal{P}\lhd\langle X\rangle_K}{\mathcal{P}|\equiv\mathcal{Q}|\sim X}$.
2) The freshness-conjuncatenation rule: $\frac{\mathcal{P}|\equiv\sharp(X)}{\mathcal{P}|\equiv\sharp(X,Y)}$.
3) The nonce-verification rule: $\frac{\mathcal{P}|\equiv\sharp(X),\mathcal{P}|\equiv\mathcal{Q}|\sim X}{\mathcal{P}|\equiv\mathcal{Q}|\equiv X}$.
4) The jurisdiction rule: $\frac{\mathcal{P}|\equiv\mathcal{Q}\Rightarrow X,\mathcal{P}|\equiv\mathcal{Q}|\equiv X}{\mathcal{P}|\equiv X}$, $\frac{\mathcal{P}|\equiv(X,Y)}{\mathcal{P}|\equiv X}$, $\frac{\mathcal{P}\lhd(X,Y)}{\mathcal{P}\lhd X}$, $\frac{\mathcal{P}|\equiv\mathcal{Q}|\sim(X,Y)}{\mathcal{P}|\equiv\mathcal{Q}|\sim X}$.

In the following, we set out the authentication goals for our proposed scheme which intends to share a mutually agreed session key.

1) **Goal 1**: $U_i \mid\equiv (U_i \xleftrightarrow{SK} S_j)$
2) **Goal 2**: $S_j \mid\equiv (U_i \xleftrightarrow{SK} S_j)$

The idealized message sequences corresponded with real multiple servers authentication protocol are detailed as follows.

1) Message m_1: $U_i \rightarrow S_j$: $(\{RID_i, \langle SID_j\rangle_{K_i}\}_{\langle X'\rangle}, X)$
2) Message m_2: $S_j \rightarrow RC$: $(SID_j, \{\langle X, Y, SID_j\rangle_{s_j}, \{RID_i, \langle SID_j\rangle_{K_i}\}_{\langle X'\rangle}, X, \quad Y\}_{s_j})$
3) Message m_3: $RC \rightarrow S_j$: $\{Y', \{SK_{ij}, Y, (S_j \mid\sim Y)\}_{K_i}, (U_i \overset{SK_{ij}}{\rightleftharpoons} S_j), SK_{ij}\}_{r_j}$
4) Message m_4: $S_j \rightarrow U_i$: $\{SK_{ij}, X, Y, (S_j \mid\sim Y)\}_{K_i}$
5) Message m_5: $U_i \rightarrow S_j$: $\langle SID_j, X, Y\rangle_{SK_{ij}}$

For validating our scheme, we list the following assumptions which are apparent and necessary.

- Assumption a_1: $U_i \mid\equiv (U_i \xleftrightarrow{K_i} RC)$
- Assumption a_2: $S_j \mid\equiv (S_j \xleftrightarrow{s_j} RC)$
- Assumption a_3: $RC \mid\equiv (S_j \xleftrightarrow{s_j} RC)$
- Assumption a_4: $U_i \mid\equiv \sharp(X)$
- Assumption a_5: $S_j \mid\equiv \sharp(Y')$
- Assumption a_6: $S_j \mid\equiv RC \Rightarrow (Y', \{SK_{ij}, Y, (S_j \mid\sim Y)\}_{K_i}, (U_i \overset{SK_{ij}}{\rightleftharpoons} S_j), SK_{ij})$
- Assumption a_7: $S_j \mid\equiv \beta$
- Assumption a_8: $U_i \mid\equiv RC \Rightarrow (SK_{ij}, X, Y, (S_j \mid\sim Y))$
- Assumption a_9: $U_i \mid\equiv \alpha$
- Assumption a_{10}: $S_j \mid\equiv U_i \Rightarrow (SID_j, X, Y)$

From the above assumptions and the idealized message sequences of our protocol, we could proceed to demonstrate that U_i and S_j share a mutually agreed session key eventually.

RC receives m_2, then we can prove:

$$RC \lhd (SID_j, \{\langle X, Y, SID_j\rangle_{s_j}, \{RID_i, \langle SID_j\rangle_{K_i}\}_{\langle X'\rangle}, X, Y\}_{s_j}).$$

From the jurisdiction rule, we can prove:

$$RC \lhd \{\langle X, Y, SID_j\rangle_{s_j}, \{RID_i, \langle SID_j\rangle_{K_i}\}_{\langle X'\rangle}, X, Y\}_{s_j}.$$

According to the assumption a_3 and the message-meaning rule, we can prove:

$$RC \mid\equiv S_j \mid\sim (\langle X, Y, SID_j\rangle_{s_j}, \{RID_i, \langle SID_j\rangle_{K_i}\}_{\langle X'\rangle}, X, Y).$$

From the jurisdiction rule, we can prove:

$$RC \mid\equiv S_j \mid\sim Y.$$

S_j receives the message m_3, we can prove:

$$S_j \lhd \{Y', \{SK_{ij}, Y, (S_j \mid\sim Y)\}_{K_i}, (U_i \overset{SK_{ij}}{\rightleftharpoons} S_j), SK_{ij}\}_{r_j}.$$

According to the assumption a_2 and the message-meaning rule, we can prove:

$$S_j \mid\equiv RC \mid\sim (Y', \{SK_{ij}, Y, (S_j \mid\sim Y)\}_{K_i}, (U_i \overset{SK_{ij}}{\rightleftharpoons} S_j), SK_{ij}).$$

According to the assumption a_5 and the freshness-conjuncatenation rule, we can prove:

$$S_j \mid\equiv \sharp(Y', \{SK_{ij}, Y, (S_j \mid\sim Y)\}_{K_i}, (U_i \overset{SK_{ij}}{\rightleftharpoons} S_j), SK_{ij}).$$

From the obtained conclusion $S_j \mid\equiv RC \mid\sim (Y', \{SK_{ij}, Y, (S_j \mid\sim Y)\}_{K_i}, (U_i \overset{SK_{ij}}{\rightleftharpoons} S_j), SK_{ij})$ and the nonce-verification rule, we can prove:

$$S_j \mid\equiv RC \mid\equiv (Y', \{SK_{ij}, Y, (S_j \mid\sim Y)\}_{K_i}, (U_i \overset{SK_{ij}}{\rightleftharpoons} S_j), SK_{ij}).$$

According to the assumption a_6 and the jurisdiction rule, we can prove:

$$S_j \mid\equiv (Y', \{SK_{ij}, Y, (S_j \mid\sim Y)\}_{K_i}, (U_i \overset{SK_{ij}}{\rightleftharpoons} S_j), SK_{ij}).$$

From the jurisdiction rule, we can prove:

$$S_j \mid\equiv (U_i \overset{SK_{ij}}{\rightleftharpoons} S_j).$$

U_i receives message m_4, we can prove:

$$U_i \lhd \{SK_{ij}, X, Y, (S_j \mid\sim Y)\}_{K_i}.$$

According to the assumption a_1 and the message-meaning rule, we can prove:

$$U_i \mid\equiv RC \mid\sim (SK_{ij}, X, Y, (S_j \mid\sim Y)).$$

According to the assumption a_4 and the freshness-conjuncatenation rule, we can prove:

$$U_i \mid\equiv \sharp(SK_{ij}, X, Y, (S_j \mid\sim Y)).$$

From the obtained conclusion $U_i \mid\equiv RC \mid\sim (SK_{ij}, X, Y, (S_j \mid\sim Y))$ and the nonce-verification rule, we can prove:

$$U_i \mid\equiv RC \mid\equiv (SK_{ij}, X, Y, (S_j \mid\sim Y)).$$

According to the assumption a_8 and the jurisdiction rule, we can prove:

$$U_i \mid\equiv (SK_{ij}, X, Y, (S_j \mid\sim Y)).$$

From the jurisdiction rule, we can prove:

$$U_i \mid\equiv (S_j \mid\sim Y),$$
$$U_i \mid\equiv Y.$$

Due to the session key $SK = \alpha \times Y = \alpha \times \beta \times P$ and the assumption a_9, we can prove:

$$U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j) \quad \textbf{(Goal 1)}.$$

S_j receives the message m_5, we can prove:

$$S_j \triangleleft \langle SID_j, X, Y\rangle_{SK_{ij}}.$$

From the obtained conclusion $S_j \mid\equiv (U_i \overset{SK_{ij}}{\rightleftharpoons} S_j)$ and message-meaning rule, we can prove:

$$S_j \mid\equiv U_i \mid\sim (SID_j, X, Y).$$

According to the assumption a_5 and the freshness-conjuncatenation rule, we can prove:

$$S_j \mid\equiv \sharp(SID_j, X, Y).$$

From the obtained conclusion $S_j \mid\equiv U_i \mid\sim (SID_j, X, Y)$ and the nonce-verific-ation rule, we can prove:

$$S_j \mid\equiv U_i \mid\equiv (SID_j, X, Y).$$

According to the assumption a_{10} and the jurisdiction rule, we can prove:

$$S_j \mid\equiv (SID_j, X, Y).$$

From the jurisdiction rule, we can prove:

$$S_j \mid\equiv X.$$

From the conclusion $S_j \mid\equiv U_i \mid\sim (SID_j, X, Y)$ and the jurisdiction rule, we can prove:

$$S_j \mid\equiv U_i \mid\sim X.$$

Due to the session key $SK = \beta \times X = \alpha \times \beta \times P$ and the assumption a_7, we can prove:

$$S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j) \quad \textbf{(Goal 2)}.$$

B. DISCUSSION ON POSSIBLE ATTACKS

In this section, we present the informal security analysis to show that our proposed scheme is equipped to preserve beneficial functionality and security properties, and also to resist a range of known network attacks.

1) PRESERVING USER PRIVACY

Unlike Jangirala et al.'s proposal, we replace the real username with a blind one RID_i as the login element, which is encapsulated in CID_i with $H(\alpha \times Q)$. Only RC can compute $X' = x \times X$ with the primary secret key x and retrieve it by decrypting $CID_i = E_{H(X')}(RID_i, H(K_i \| SID_j)) = E_{H(\alpha\times Q)}(RID_i, H(K_i \| SID_j))$. The approach for the adversary to acquire RID_i is infeasible, because it is extremely difficult for him/her to manage the computational Diffie-Hellman problem (computing $X' = \alpha \times Q$ from $X = \alpha \times P, Q = x \times P$). On the other hand, there exists no invariable value in the login request message $\{CID_i, X\}$, which are both invoked by random number α. Thereby, the adversary can not identify and trace the user who participate in the login session. And hence, the proposed scheme achieves user anonymity and untraceability protection.

2) OFF-LINE PASSWORD GUESSING ATTACK

Many research papers have demonstrated that portable devices can be breached by a logical or physical approach [3], [4]. Herein, the adversary is endowed with the ability that compromises the credential stored in the portable devices. If he/she extracts $\{A_i, B_i, r\}$ from the victim user U_i's portable device, where $B_i = H(A_i \oplus H(ID_i \| H(PW_i \| r)))$, the adversary tries to crack the invoked password. From the equation we can see, U_i's username ID_i and password PW_i are the key values to establish the verification. It means that the adversary has to seek out the matched ID_i and PW_i simultaneously. The infeasibility of guessing two parameters correctly synchronously in polynomial time shows that our scheme could resist this attack with portable device breach.

3) IMPERSONATION ATTACK

The impersonation attack is an attack launched by the adversary for masquerading validated users and illegitimately accessing information services. The prerequisite for performing the attack is to forge a login request $\{CID_i, X\}$ which can successfully be verified by S_j and RC, where $CID_i = E_{H(X')}(RID_i, H(K_i \| SID_j)) = E_{H(\alpha\times Q)}(RID_i, H(K_i \| SID_j))$. U_i's blind username RID_i and the matched secret key K_i are the core for generating the login request. In the above analysis, we have demonstrated that our proposed scheme can achieve username and password confidentiality, and thus the safeguarded K_i by these two values is also secure. Thereby, the impersonation attack is infeasible in the proposed scheme.

4) SERVER SPOOFING ATTACK AND REGISTRATION CENTER SPOOFING ATTACK

In the server spoofing attack, if a legal but malicious user of the system tries masquerade as the legitimate service providing server S_j, he/she needs to forge a valid reply message $\{C_i\}$ to U_i. However, since the adversary cannot recover it from D_j without the knowledge of s_j, he/she also can not generate it without knowing secret values of U_i. If the attacker is a legal but malicious server of the system, he/she also cannot masquerade as another server to deceive any legal user due to he/she cannot obtain the secret keys of other servers to generate the correct response message. Hence, server spoofing attack is meaningless in our scheme.

In the registration center spoofing attack, any participant of the system cannot get the master secret key x, which is held privately by the registration center only. Therefore, any attacker cannot masquerade as the registration center.

TABLE 2. Comparisons of functionality.

	Guo et al.'s [15]	Kumari et al.'s [27]	Jangirala et al.'s [28]	Amin et al.'s [29]	Ours
Resistance of impersonation attack	Yes	No	No	No	Yes
Resistance of off-line password guessing attack	No	Yes	Yes	Yes	Yes
Resistance of server spoofing attack	Yes	No	No	No	Yes
Safeguarding user anonymity	Yes	No	No	No	Yes
Resistance of replay attack	No	Yes	Yes	Yes	Yes
Providing known key security	Yes	Yes	Yes	No	Yes
Providing perfect forward secrecy	Yes	Yes	Yes	No	Yes

5) REPLAY ATTACK

Replay attacks can be prevented by tagging each message with a session varied component number. In our scheme, the transmitted messages are all contained with a random nonce. Even if the adversary replays any authentication messages exchanged between U_i, S_j, RC, the receivers can easily detect the replayed one in the verification phase.

6) FORWARD SECRECY

The forward secrecy is an outstanding property of information exchange protocol, which means that compromise of long-term keys does not reveal past session key. In our scheme, U_i and S_j agree the session key $SK = \alpha \times \beta \times P$ with random numbers $\{\alpha, \beta\}$ generated by user and server, separately. These two values are only accessible for user and server, and anyone has to computes SK from $X = \alpha \times P$, $Y = \beta \times P$. In other words, the secrecy of session key is reduced to the intractability of computation Diffie-Hellman problem. Consequently, the leakage of primary secret key x of RC does not compromise the current session key.

7) KNOWN KEY ATTACK

The known key attack is an attack model that adversaries utilize one leaked session key to compromise prior sessions. In our scheme, the session key $SK = \alpha \times \beta \times P$ is computed by two session varied random nonces α and β. The calculation method guarantees that each session key SK is independent of the other one. Therefore, the leakage of SK does not affect other unexposed sessions.

VI. PERFORMANCE AND FUNCTIONALITY ANALYSIS

In this segment, we summarize the evaluation of computation overhead and security functionality, and make comparisons with other related solutions [15], [27]–[29]. It is visible from Table 2 that our scheme can thwart a range of security threats which the other schemes suffers from and provide stronger security and better usability than the other schemes. In addition, we also provide the formal proof of BAN-logic to reason that our proposed scheme achieves tripartite authentication.

In Fig 5 and Fig 6, we illustrate the comparisons of computational consumption in the client-side and server & registration center, respectively. As shown in Fig 5 and Fig 6, the computational costs for both client side and server & registration center side are lower than those of compared schemes except Jangirala *et al.*'s scheme [28] and Amin *et al.*'s scheme [29]. Especially, these two schemes are susceptible to some malicious attacks listed in Table 2.

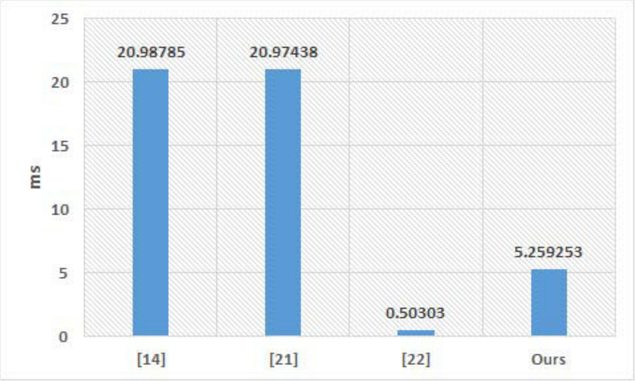

FIGURE 5. Computation consumption in client for one round of authentication.

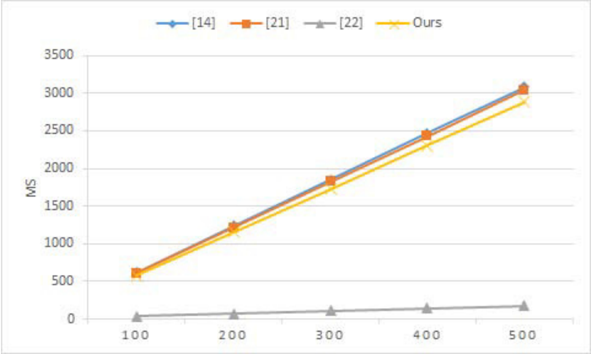

FIGURE 6. Computation consumption in servers & registration center.

The simulation of server & registration center is implemented at Python 3.5.2 using an Intel(R) Core(TM) i5-4590 CPU @ 3.30GHZ with 3300MB RAM in Ubuntu 16.04 system. We simulate portable device & smart card of users at Python 3.5.2 using an Intel(R) Core(TM) i5-4590 CPU at 1.65GHZ with 1540MB RAM in Ubuntu 16.04 system. We employ SHA-256, advanced encryption standard (AES) to implement hash function, symmetric encryption/decryption algorithm. Furthermore, the modular exponentiation algorithm and point multiplication operation is executed in the multiplication cycle group with 1024-bit security parameter and MNT asymmetric group (MNT224), separately.

VII. CONCLUSION

In this article, we identified several vulnerabilities in Jangirala et al.'s authentication scheme for multiple servers authentication architecture. All registered users and servers share a common secret value $H(y)$ and it is the fatal issue that results in the aforementioned security flaws. Hence, we proposed an enhanced scheme with a tripartite certification strategy. Noticeably, user's secret key overhead in our scheme is $O(1)$ and it is more applicable in the wireless mobile environment. Besides, we present the formal proof to analyze the proposed scheme by employing BAN-logic. It is better to reduce the security of the scheme to some mathematical problems which are believed hard in a standard model or random oracle model. How to devise a secure multiple servers authentication and key agreement protocol in the standard model or random oracle model is an interesting topic for our future work.

ACKNOWLEDGMENT

The authors are grateful to the editor and anonymous reviewers for their valuable suggestions.

REFERENCES

[1] L. Zhou, V. Varadharajan, and M. Hitchens, "Achieving secure role-based access control on encrypted data in cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 12, pp. 1947–1960, Dec. 2013.

[2] J. Xu, Q. Wen, W. Li, and Z. Jin, "Circuit ciphertext-policy attribute-based hybrid encryption with verifiable delegation in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 119–129, Jan. 2016.

[3] P. Kocher and J. B. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Annu. Int. Cryptol. Conf.*, Aug. 1999, pp. 388–397.

[4] T. S. Messerges, E. R. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[5] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Heidelberg, Germany: Springer, 2003.

[6] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.

[7] L.-H. Li, I.-C. Lin, and M.-S. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1498–1504, Nov. 2001.

[8] R. Admin, "Cryptanalysis of efficient dynamic ID based remote user authentication scheme in multi-server environment using smart card," *Int. J. Netw. Secur.*, vol. 18, no. 1, pp. 172–181, 2016.

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

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

[11] S. K. Sood, A. K. Sarje, and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," *J. Netw. Comput. Appl.*, vol. 34, no. 2, pp. 609–618, 2011.

[12] 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," *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 763–769, 2012.

[13] 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 Syst. Appl.*, vol. 38, no. 11, pp. 13863–13870, 2011.

[14] X. Li, J. Ma, W. Wang, Y. Xiong, and J. Zhang, "A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments," *Math. Comput. Model.*, vol. 58, nos. 1–2, pp. 85–95, 2013.

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

[16] B. Wang and M. Ma, "A smart card based efficient and secured multi-server authentication scheme," *Wireless Pers. Commun.*, vol. 68, no. 2, pp. 361–378, 2013.

[17] R. S. Pippal, C. D. Jaidhar, and S. Tapaswi, "Robust smart card authentication scheme for multi-server architecture," *Wireless Pers. Commun.*, vol. 72, no. 1, pp. 729–745, Sep. 2013.

[18] J.-L. Tsai, N. W. Lo, and T.-C. Wu, "A new password-based multi-server authentication scheme robust to password guessing attacks," *Wireless Pers. Commun.*, vol. 71, no. 3, pp. 1977–1988, Aug. 2013.

[19] Y. Lu, L. Li, H. Peng, and Y. Yang, "Cryptanalysis and improvement of a chaotic maps-based anonymous authenticated key agreement protocol for multiserver architecture," *Secur. Commun. Netw.*, vol. 9, no. 11, pp. 1321–1330, Jul. 2016.

[20] R. Amin, S. H. Islam, M. K. Khan, A. Karati, D. Giri, and S. Kumari, "A two-factor RSA-based robust authentication system for multi-server environments," *Secur. Commun. Netw.*, vol. 2017, Aug. 2017, Art. no. 5989151.

[21] V. Sureshkumar, R. Amin, and R. Anitha, "An enhanced bilinear pairing based authenticated key agreement protocol for multiserver environment," *Int. J. Commun. Syst.*, vol. 30, no. 17, Nov. 2017, Art. no. e3358.

[22] J.-L. Tsai and N.-W. Lo, "A chaotic map-based anonymous multi-server authenticated key agreement protocol using smart card," *Int. J. Commun. Syst.*, vol. 28, no. 13, pp. 1955–1963, Sep. 2015.

[23] T. Maitra, S. H. Islam, R. Amin, D. Giri, M. K. Khan, and N. Kumar, "An enhanced multi-server authentication protocol using password and smart-card: Cryptanalysis and design," *Secur. Commun. Netw.*, vol. 9, no. 17, pp. 4615–4638, Nov. 2016.

[24] R. Amin, S. H. Islam, P. Gope, K.-K. R. Choo, and N. Tapas, "Anonymity preserving and lightweight multimedical server authentication protocol for telecare medical information system," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 4, pp. 1749–1759, Jul. 2019. doi: 10.1109/JBHI.2018.2870319.

[25] 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 Pers. Commun.*, vol. 84, no. 1, pp. 439–462, 2015.

[26] S. Shunmuganathan, R. D. Saravanan, and Y. Palanichamy, "Secure and efficient smart-card-based remote user authentication scheme for multi-server environment," *Can. J. Elect. Comput. Eng.*, vol. 38, no. 1, pp. 20–30, Mar. 2015.

[27] S. Kumari and H. Om, "Cryptanalysis and improvement of an anonymous multi-server authenticated key agreement scheme," *Wireless Pers. Commun.*, vol. 96, no. 2, pp. 2513–2537, Sep. 2017.

[28] S. Jangirala, S. Mukhopadhyay, and A. K. Das, "A multi-server environment with secure and efficient remote user authentication scheme based on dynamic ID using smart cards," *Wireless Pers. Commun.*, vol. 95, no. 3, pp. 2735–2767, Aug. 2017.

[29] R. Amin, S. H. Islam, M. S. Obaidat, G. P. Biswas, and K.-F. Hsiao, "An anonymous and robust multi-server authentication protocol using multiple registration servers," *Int. J. Commun. Syst.*, vol. 30, no. 18, Dec. 2017, Art. no. e3457.

HUAWEI WANG received the bachelor's degree from Xinjiang University, in 2013. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Networking and Switching Technology, Network Security Research Center, Beijing University of Posts and Telecommunications (BUPT). He is also doing research with the Institute of Network and Technology, BUPT. His research interests include network security, mobile security, and blockchain technology.

DIANLI GUO received the bachelor's degree from Heze College, in 2011, the master's degree in science from the University of Jinan, Shandong, in 2014, and the Ph.D. degree in cryptology from the Beijing University of Posts and Telecommunications (BUPT), in 2018. He is currently doing research with The 6th Research Institute of China Electronics Corporation. His research interests include network security and cryptography protocols.

QIAOYAN WEN received the B.S. and M.S. degrees in mathematics from Shaanxi Normal University, Xi'an, Shaanxi, China, in 1981 and 1984, respectively, and the Ph.D. degree in cryptography from Xidian University, Xi'an, in 1997. She is currently a Professor with the Beijing University of Posts and Telecommunications. Her current research interests include coding theory, cryptography, information security, the Internet security, and applied mathematics.

HUA ZHANG received the B.S. and M.S. degrees from Xidian University, in 2002 and 2005, respectively, and the Ph.D. degree in cryptology from the Beijing University of Posts and Telecommunications (BUPT), in 2008, where she is currently an Associate Professor with the Institute of Network Technology. Her research interests include cryptographic protocols, cloud computing security, and industrial control systems security.