Secure and Lightweight Trustee-based Scheme to Establish Mutual Authentication in Cloud Computing Environments

Mr.Sabout Nagaraju,  
Assistant Professor, Department of Computer Science  
Pondicherry University Community College  
Lawspet, Pondicherry-605 008, India

Dr.S.K.V. Jayakumar  
Assistant Professor, Department of Computer Science  
School of Engineering & Technology, Pondicherry University  
Kalapet, Pondicherry-605 014, India

Abstract  
Due to expeditious advancement in the cloud computing architecture, stakeholders are migrating and relaying on different third party cloud service providers. Moreover, the user applications, data and processes are running out of the premises. Thus wherever it is running on thirty party’s environment, the security becomes an issue. Especially, public cloud environment mandates savior security and access control mechanisms to safeguard the cloud services and its outsourced assets in a seamless and transparent manner. An effective authentication is the basis, topmost prioritized and emergence one for the secure cloud communications. We reviewed, analyzed and found that recently proposed schemes are insecure against the service provider impersonation and Ephemeral Secret Leakage (ESL) attacks and also not able to support mutual authentication. As a result, in this article a secure and lightweight mutual authentication scheme is proposed based on tokenization and time-based dynamic nonce generation. Our investigation restricts the external and internal malicious users from accessing cloud-based critical information. The proposed protocol not only meets intended mutual authentication, but also provides security strength against the impersonation and ephemeral secret leakage attacks.

Keywords—Mutual Authentication, Single Sign-On, Elliptic-Curve, Cloud Service Provider, Identity Provider, Trustee

I. INTRODUCTION

An illusion of the computing has gone through different phases, staring from the simple system to mini, mainframe computers and then evolved to clustered, grid and distributed computing and so on. In last one decade, it is observed that there is a need of economical storage, processing power, software and hardware resources as a service are on demand. These have become possible due to advancement in the cloud computing services. In today’s life, cloud computing is becoming a part and parcel of our life. The expeditious development in Internet-of-Things, big data, mobile and social networks require cloud computing to provide economical data storage and high-speed computing capabilities. However, these imperatives are rapidly emerging as pillars for the smarter daily life and official works [1]. As per 2018 Global Cloud Data Security Study report [2], IT professionals estimated that 79% of the today’s business operations are using cloud computing services and it may be expected to rise to 87% in next two years. Importantly, IT experts also reported that 39% of the organizations are using cloud computing services for their total IT and data processing needs and it may be expected to rise to 51% in next two years. And IT executives observed that on average, 43% of all corporate data is stored in the cloud data centers. So cloud computing is basically a internet-based resource enabling technology, which allows the users to work on scale up and scale down models with pay-per-usage policy [3]. Mutual authentication is an ever-growing need process to verify entities in an internet-based communication technology and authenticate each other genuinely. However, the mutual authentication solutions designed for on-premises will not be working for the cloud computing systems. On the other hand, for some financial/personal gain, a cloud rogue system administrator may steal the user authentication credentials or alter user privileges for acquiring sensitive information [4]. Similarly, dissatisfied cloud staff may exfiltrate sensitive information. For maintaining trust and reputation, dishonest cloud service providers may hide the data loss; instead they may return the stale data. Dishonest CSPs may completely deny the service requests or execute few requests in order to save computing resources.

To overcome the above problems, Jia-Lun Tsai et al.[5] and Debiao He et al. [6] schemes mostly being used is the bilinear key pairing cryptosystem with dynamic nonce generation to support mutual authentication, key exchange, user anonymity, and user untraceability. Their schemes cannot resist the service provider impersonation attack, i.e., an adversary can impersonate the service provider to the user. Also, the adversary can extract the user’s real identity, private key and session key materials during the authentication process are called ephemeral secret leakage attack. The authors also do not take into account following considerations:
First, single sign-on (SSO) authentication requirement, since the users may access different kind of cloud computing services from the multiple service providers. However, most of the single sign-on authentication approaches require a trustee participation and trustee could become a bottleneck for the user authentication.

Second, mutual authentication using tokenization and time-based dynamic nonce generation and establishment, since the cloud computing environment is insecure and there is a chance of ephemeral secret leakage attack.

Third, the computation cost need to be minimized, since cloud computing is emerging with mobile technology and internet of things in which smart phones and sensors have limited computing capacity.

Fourth, very critical and barely explored issue that should be taken into consideration is rogue/dissatisfied/dishonest/untrusted CSP’s.

This paper further divided into seven sections. Literature reviews are presented in Section 2. System-level model and assumptions of our trusted authentication scheme is illustrated in Section 3. Section 4 describes our investigation. Section 5 discusses completeness of the proposed authentication scheme. Section 6 reports the security and performance evaluation. Section 7 summarizes the proposed trustee-based authentication scheme.

II. RELATED WORK

Over the last more than three and half decades, several mutual authentication mechanisms have been presented to authenticate each other, in order to protect critical data. To bring stronger security in the mutual authentication Lamport [6] proposed first mutual authentication scheme for insecure open networks using sequence of hashed passwords. This scheme not able to resist the replay attack and the impersonation attacks. To improved security in the mutual authentication, several schemes [7-11] are proposed with different hash collision resistant functions and elliptic curves. In these schemes each entity has to maintain a verifier table to achieve the mutual authentication. The adversary may impersonate the user or the server when attacker steals the verifier tables. Besides, these schemes suffer from the denial of service attack if the adversary modifies the verifier table maliciously. To remove above schemes weaknesses, more securable approaches [12-17] are invented without any verifier table. These schemes provide mutual authentication by using both the password and the smart card and no verifier table is needed. Schemes supports better security in mutual authentication and gives better performance. Observing limitations, these schemes designed for on-premises server applications and cannot be directly used in cloud computing environment because many CSP exist and the user has to register in every CSP repeatedly. From security reasons, the user not only has to put extra efforts in remembering many different passwords and identities, but also wastes a lot of time to execute repeated registration. It would be easier for the user, if all the applications have a common user credentials database. Here, user can access all the applications using one set of login credentials. This type of authentication is called single sign-on (SSO) authentication.

To achieve SSO-based mutual authentication, several schemes [13-27] have been using dynamic pseudonym IDs, password and the biometric-based smart cards. Observing limitations, the performance of the schemes are not acceptable because of complicated neural networks and discrete logarithm computations are used to implement secure mutual authentication. These schemes are insecure against the impersonation attack and not able to provide user anonymization. These schemes also cannot support the perfect forward secrecy. To overcome the problems of above schemes, Jia-Lun Tsai et al.[5] and Debiao He et al. [6] schemes mostly being used is the bilinear key pairing cryptosystem with dynamic nonce generation to support mutual authentication, key exchange, user anonymity, and user untraceability. This scheme cannot resist the service provider impersonation attack, i.e., an adversary can impersonate the service provider to the user. Also, the adversary can extract the user’s real identity, private key and session key materials during the authentication process are called ephemeral secret leakage attack.

III. SYSTEM-LEVEL FRAMEWORK AND ASSUMPTIONS

In this section a typical high-level architecture is presented for distributed cloud computing environment which consists of set of cloud service providers, different identity providers and users as shown in Figure 1. The personal and sensitive information of the data owner or enterprise will be managed in geographically distributed different cloud service provider’s data centers. The cloud service providers outsource cheap, flexible and on-demand storage space and computing capabilities to the data owner to make this information available any time to the legitimated users. Trusted Identity Provider (TIdP) is a set of distributed servers that are managed by an organization or board of eminent security researchers. It is a separation from the cloud computing resource servers and will run on a separate trusted lockdowns security platforms. Trusted identity provider replaces the sensitive authentication parameters with tokens or surrogate values and also maps back to the real data by making use of a secure enclave or look-up table’s. A cloud resource server validates the identity and access tokens generated by the identity provider, if are valid, then user will be directed to access cloud application. Trusted identity provider protects sensitive authentication parameters and access tokens and are never been revealed to the service
providers. Trusted identity provider also support to compute session key materials and generates identity and access tokens for the user authentication. In our proposed scheme, legitimated users and CSPs must rely on an identity provider. The notations and their meanings we used for describing our framework are listed in Table 1.

Table 1. List of Abbreviations

<table>
<thead>
<tr>
<th>Notations</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{pa}()$</td>
<td>A public-key encryption function</td>
</tr>
<tr>
<td>$D_{pa}()$</td>
<td>A decryption function’s corresponding to $E_{pa}()$</td>
</tr>
<tr>
<td>$e()$</td>
<td>A symmetric encryption’s function</td>
</tr>
<tr>
<td>$d_{k}()$</td>
<td>A symmetric decryption’s function corresponding to $e_{k}()$</td>
</tr>
<tr>
<td>$\hat{e}$</td>
<td>Exponential function</td>
</tr>
<tr>
<td>$U^i$</td>
<td>The ID which $U^i$ inputs in authentication phase</td>
</tr>
<tr>
<td>$pwd_{U^i}$</td>
<td>The password which $U^i$ inputs in authentication phase</td>
</tr>
<tr>
<td>$MAC_{U^i}$</td>
<td>User MAC address which submits in authentication phase</td>
</tr>
<tr>
<td>$U^i$</td>
<td>The ID which $U^i$ inputs in registration phase</td>
</tr>
<tr>
<td>$pwd_{U^i}$</td>
<td>The password which $U^i$ inputs in registration phase</td>
</tr>
<tr>
<td>$MAC_{U^i}$</td>
<td>User MAC address submitted in registration phase</td>
</tr>
<tr>
<td>$M_i$</td>
<td>The message sent by various communication entities.</td>
</tr>
<tr>
<td>$ToT_o$</td>
<td>Ex-OR operation time and One-way Hash operation time</td>
</tr>
<tr>
<td>$AT&amp;RT$</td>
<td>Authentication Type and Requested Authentication Type</td>
</tr>
<tr>
<td>$CSPTID_j$</td>
<td>The Service Provider ID which inputs in the registration phase</td>
</tr>
<tr>
<td>$CSPID_j$</td>
<td>The Service Provider ID which CSPs inputs in the authentication phase</td>
</tr>
<tr>
<td>$h_i()$</td>
<td>$i^{th}$ one-way hash function</td>
</tr>
<tr>
<td>$</td>
<td></td>
</tr>
<tr>
<td>$salt$</td>
<td>A random data that is used in generating a hashed password and also avoids the hash collisions.</td>
</tr>
<tr>
<td>+K &amp; -K</td>
<td>Public keys and Private keys</td>
</tr>
<tr>
<td>FPR&amp;FNR</td>
<td>False Positive Rate and False Negative Rate</td>
</tr>
</tbody>
</table>

IV. TRUSTED AUTHENTICATION PROTOCOL

In this section we describe a trustee-based mutual authentication scheme. Mutual authentication is an imperative security process for recognizing genuine users by authenticating each other. Proposed authentication is designed with the combination of User ID and Password (what you know), MAC address (where you are) and OTP (what you have). Proposed mechanism provides the ability for a user to use same set of credentials to logon to multiple applications of different CSPs. The identity and access tokens generated by the identity provider and received from the user will be just matched in the cloud servers. The control flow of the proposed authentication entities is represented in Figure 2. This approach helps the users to protect identity and access management tokens from the malicious insiders and other external unauthorized adversaries.

The authentication scheme has three phases as follow.

Initialization phase, First, Trusted Identity Provider ($TIdP$) chooses a random number as private key ($k_{TIdP}$) and computes $P_{TIdP} = h_1(P_{TIdP})$ as its corresponding public key, where $h_1$ is a one-way hashing function. Likewise, CSP chooses a random number as a private key ($k_{CSP_j}$) and computes $P_{CSP_j} = h_2(P_{CSP_j})$ as its corresponding public key, where $h_2$ is a one-way hashing function and publishes $P_{CSP_j}$ and its service attributes.

Registration phase, Each User ($U^i$) filters CSPs based on service attributes like $\{STD Ts, SS\geq DS, PS\geq RS, SC\leq SP\}$, where $ST$ indicates storage type, $DT$ presents data type, $SS$ indicates storage space, $DS$ represents data space, $PS$ indicates processing speed, $RS$ represents requested speed, $SC$ indicates service cost and $SP$ represents service pay. User ($U^i$) sends his/her chosen $CSPID_j$, user-id ($ID_{U^i}$), password ($pwd_{U^i}$) and device MAC address ($MAC_{U^i}$) to $IdP$ for registration.
Where, each user selects a desired cloud service provider (CSP) based on the global trust evaluation mechanism which is described in [26]. Identity provider computes $h_2(password + salt) = HPWD$, $h_4(\text{MAC}(U_i)) = h_4$ ($\text{MAC}(U_i) = \text{HMAC}$, where $h_2(.)$ and $h_4(.)$ are the one-way hashing functions and stores these values in distributed and highly secured databases. IdP sends $ID_{Ui}$ and mutual operation on nonce to user through secure channel. Similarly, cloud service provider also registers with IdP.

**The authentication phase** performs the following steps to validate remote user ($U_i$) login credentials by authenticating each other.

1. **User $U_i$** inputs $ID_{Ui}^*$ and $CSP_{Ui}^*$ and performs public key encryption on concatenation of $ID_{Ui}^*$, $CSP_{Ui}^*$ and $n_1$ and computes message as $M_1 = E_{pkCSP_{j}}(ID_{Ui}^* || CSP_{Ui}^*) || n_1$ and then sends $M_1$ to $TldP$.

2. **$TldP$** decrypts user message $M_1$ details such as $ID_{Ui}^*$, $CSP_{Ui}^*$ and $n_1$ by using his/her private key $PrCSP_{j}$. And then check for $ID_{Ui}^* = ID_{Ui}$ and $CSP_{Ui}^* = CSP_{Ui}$. If matches, then $TldP$ performs a mutual authentication operation on nonce as $n_2 = n_2>>1 \mod n$ and then derives $M_2 = e_{HPWD}(RN)) || n_1$ using user hashed password $pwd_{Ui}$. $TldP$ sends $M_2$ to the user. If $ID_{Ui}^*$ or $CSP_{Ui}^*$ is not found or invalid, then the user request will be rejected.

3. **User extracts $n_2$ from message $M_2$ and then checks for mutual authentication value i.e., $n_2 = n_2>>1 \mod n$. If nonce value is matched then user decrypts secrete one-time random number $RN$ using his/her hashed password $pwd_{Ui}$ i.e., $RN = d_{HPWD}(e_{HPWD}(RN))$. Next, user computes a message $M_3 = e_{\text{HMAC}}(RN || NA_i) || n_3$ and sends $M_3$ to the $TldP$, where $NA_i$ is the user network address details and $n_3 = n_3>>1 \mod n$.

4. **$TldP$** obtains $RN$ by decrypting user message $M_3$ using user registered HMAC and then check for $RN^* = RN$. If matches, then $TldP$ perform tokenization of OTP and $NA_i$ using advanced tokenization techniques proposed by Sabout N. et al. [28]. Tokenized one-time password $OTP_{T}$ is sends to the user registered mobile number and then computes $n_4 = n_4>>1 \mod n$ and $Token_{Ui} = E_{pkCSP_{j}}(OTP_{T} || NA_i)$ and then derives $M_4 = Token_{Ui} || n_4$. $TldP$ sends $M_4$ to the user module and redirects to submit tokenized details to the $CSP_{j}$. If $RN^* = RN$ is not found or invalid, then the user request will be rejected.

5. **User obtains $n_4$ from $M_4$ and then checks for mutual authentication value i.e., $n_4 = n_4>>1 \mod n$, if it matches, then user is allowed to enter OTP and derives $M_5 = (CSP_{Ui}^* || Token_{Ui} || n_4) || OTP_{T}$ and then sends $M_5$ to $CSP_{j}$. If $n_5 = n_5>>1 \mod n$, then the authentication process will be terminated.

6. From $M_5$, $CSP_{j}$ extracts $CSP_{Ui}^*$, $Token_{Ui}$, $OTP_{T}$ and $NA_i^*$ and then decrypts $Token_{Ui}$ by using private key as $D_{pkCSP_{j}}(Token_{Ui})$. Next, checks for $CSP_{Ui}^* = CSP_{Ui}$, if matches, then proceeds with further communication.

7. **Next, $CSP_{j}$ requests corresponding user identity and access details from $TldP$ and then checks received and requested $OTP_{T}$ and $NA_i^*$ details. If matches, then user is allowed to use cloud service application. And then derives with $M_6 = (n_6)$ and then sends $M_6$ to $U_i$. If token details are not

**Fig.2. Control Flow of the Proposed Mutual Authentication Scheme**
Let the basic terminologies are introduced on protocol security by using suitable postulates as follows.

V. COMPLETENESS OF THE PROPOSED PROTOCOL

This section formally analyses the mutual authentication and security strength of the proposed protocol using standard GNY cryptographic logic. The analysis proved that the proposed protocol not only meets intended mutual authentication functionality, but also ensures the security strength against the service provider impersonation and other replay attacks. We used cryptographic GNY [29] belief logic to formally analyze the working nature of our trusted authentication mechanism and to verify whether our mechanism meets its goals. GNY belief logic is the substantial extension of BAN logic [30].

A. Basic Terminologies and Statements

Let CP\(_i\) be the credential parameter message and the following basic terminologies are introduced on CP\(_i\):
- \(h(CP_i)\) : hash operation on CP\(_i\)
- \(\{CP_i\}^k\) \& \(\{CP_i\}^j\) : CP\(_i\) is encrypted with +K and decrypted with -K.
- \(\{CP_i\}^k\) \& \(\{CP_i\}^{-1}\) : CP\(_i\) is encrypted and decrypted with secret key K.

**Statements:** Let \(E_i\) and \(E_j\) be two communication entities and the following statements are formed on \(E_i\) and \(E_j\):

1) \(E_i \triangleleft E_j\) : \(E_i\) holds \(E_j\)
2) \(E_j \triangleright CP\(_i\)\) : \(E_j\) possesses credential parameter message CP\(_i\)
3) \(E_i \Rightarrow CP\(_i\)\) : \(E_i\) is conveyed CP\(_i\)
4) \(E_i \equiv \#(CP\(_i\))\) : \(E_i\) believes that CP\(_i\) is fresh
5) \(E_i \equiv \phi(CE)\) : \(E_i\) believes that CP\(_i\) is recognizable
6) \(E_i \equiv E_j\) : \(E_i\) believes that S is a suitable secrete for \(E_i\) and \(E_j\)
7) \(E_i \equiv \kappa\) : \(E_i\) believes that public key +K is suitable for \(E_i\)
8) \(E_i \triangleright X\) : \(E_i\) has jurisdiction over X
9) \(E_i \triangleright X\) : \(E_i\) is told that he/she didn’t convey X previously in the current session.

B. Security Analysis

In this subsection, we describe our authentication protocol security by using suitable postulates as follows.

1) The first flow:

\[
\text{TIdP} \Rightarrow \{ID_{U,I}^*, [CP_{IDP}*]_{i-k}\} \mid n_1, \text{TIdP} \Rightarrow \#(ID_{U,I}^*)
\]

If \(U_i\) is told that the first message \(\{ID_{U,I}^*, [CP_{IDP}*]\}\) is encrypted with TIdP public key +K, then TIdP can obtains ID_{U,I}^* and CP_{IDP}* using corresponding private key -K. \(U_i\) possesses ID_{U,I}^*, CP_{IDP}* and \(n_1\) for further interactions in the current session.

2) The second flow:

\[
U_i \left(\{e_{HPWD}(RN)\} \mid n_2, U_i \triangleright HPWD, U_i \equiv \#(e_{HPWD}(RN) \mid n_2)\right) (T_3, P_1)
\]

If TIdP is being told that in second message, a random number is encrypted with \(U_i\) hashed password HPWD, then \(U_i\) obtains RN with the current session input hashed password HPWD*, TIdP possess ID_{U,I}^* and CP_{IDP}* for further interactions in the current session.

3) The third flow:

\[
\text{TIdP} \Rightarrow \{e_{HMACU}(RN) \mid n_3, TIdP \triangleright HAMCUi(\{RN\}n_1, P_1)\}
\]

If \(U_i\) is being told that the third message component i.e., \(e_{HMACU}(RN) \mid n_3\) is encrypted with the current session input hashed MAC address HAMC*, then TIdP obtains RN and NA using HAMC.

\[
\text{TIdP} \Rightarrow \{e_{HAMC}(RN) \mid n_1, \text{TIdP} \triangleright HAMCUi(\{RN\}n_1, P_1)\}
\]

Below given conditions are hold: (1) TIdP receives a message \(\{e_{HAMC}(RN) \mid n_3\}\); (2) TIdP believes that the hashed MAC address HAMC* is suitable for \(U_i\) to encrypt secret communication; (3) TIdP believes that \(U_i\) inputs HAMC*; (4) TIdP believes all the decrypted and nonsecret components are recognizable; (5) TIdP believes that \(U_i\) sent message is fresh; (6) TIdP checks for random secret RN, if it is equal, then believes that the \(U_i\) is a legitimate entity. Thus TIdP is entitled to believe that \(U_i\) is conveyed a message.

4) The fourth flow:

\[
U_i \left(\#(Token_{U,I} \mid n_4), \text{CSP} \triangleright \text{PbkCSP}_{i-k}, \text{TIdP} \triangleright \text{PbkCSP}_{i-k}\right) (T_3, P_1)
\]

If \(U_i\) is being told that in the fourth message \(\text{Token}_{U,I} \mid n_4\), \(\text{Token}_{U,I}\) is encrypted with CSP, public key \(\text{PbkCSP}_i\) and \(U_i\) can obtains \(n_4\).

\[
U_i \left(\#(\text{Token}_{U,I} \mid n_4), U_i \equiv \phi(\text{Token}_{U,I} \mid n_4), U_i \equiv \#(\text{Token}_{U,I} \mid n_4)\right)
\]

\[\text{TIdP} \triangleright \text{PbkCSP}_i \text{Token}_{U,I} \mid n_4, U_i \equiv \text{CSP} \triangleright \text{PbkCSP}_{i-k} - I_1\]
Below given conditions are hold: (1) $U_i$ receives a message $\{Token_i||n_i\}$; (2) $U_i$ believes that the CSP, public key $P_{\text{MACS}}$ is suitable for secrecte communication; (3) $U_i$ believes the message components $Token_i$ and $n_i$ are recognizable; (4) $U_i$ believes that $TIdP$ sent message is fresh; (5) $U_i$ checks $n_i$ with $n_i > l \ mod \ n$, if it is equal, then $U_i$ believes that $TIdP$ is legitimate entity.

5) The fifth flow:

\[
CSP_i \prec \{ \text{CSP}_j^* || \text{Token}_i || \text{OTP}^*_i \}, CSP_j \equiv P_{\text{MACS}}^{\text{CSP}_j} \prec \{ \text{CSP}_j^* || \text{Token}_i || \text{OTP}^*_i \} \quad (T_j, p_j)
\]

If $U_i$ is being told that in fifth flow, message $\{Token_i||n_i\}$ is encrypted with CSP$_j$ public key $P_{\text{MACS}}$, then CSP$_j$ obtains $Token_i$ using private key $P_{\text{MACS}}$. CSP$_j$ decrypts $\{Token_i||n_i||\text{OTP}^*_i\}$, CSP$_j$ \(\equiv \phi (\text{CSP}_j^* || \text{Token}_i || \text{OTP}^*_i)\). CSP$_j$ accepts $\{Token_i||n_i||\text{OTP}^*_i\}$, if $\phi(\text{Token}_i||\text{OTP}^*_i) \equiv \text{P}_{\text{MACS}}^j$, then $U_i$ accepts the message is authentic.

By following conditions are hold: (1) CSP$_j$ receives a message $\{Token_i||n_i||\text{OTP}^*_i\}$ in which $\text{Token}_i$ is encrypted with CSP$_j$ public key $P_{\text{MACS}}$, then CSP$_j$ believes that $P_{\text{MACS}}$ is a secrete private key; (2) CSP$_i$ verifies that obtained session materials are recognizable; (3) CSP$_i$ accepts the received message is fresh; (5) CSP$_i$ sends a request for acquiring identity and access tokens from $TIdP$. (6) CSP$_i$ checks identity and access tokens received from user and $TIdP$, if matches, then CSP$_i$ believe that the user is legitimate entity and user is allowed to use the cloud service.

VI. PERFORMANCE EVALUATION

The proposed algorithm is evaluated by using Microsoft Azure Compute and Storage Emulator. We determine an effectiveness of the proposed protocol in terms of number of cryptographic operations, communication and computation costs required. Let $T_{bp}$ be the bilinear pairing operation time, $T_h$ indicates time required for the concatenation operation, $T_{bp}$ is one-way hash operation time, $T_c$ represents time required for Exclusive-OR operation and $T_i$ and $T_m$ are the inverse and additive multipication operation time respectively. The comparison study of the computation cost and their related operations running times are listed in Table 2 and 3. In general, concatenation and bitwise Exclusive-OR operations are much faster and will consume constant timings, so that these two operations time can be neglected in calculating computation cost. Therefore, for registration process, our scheme requires two hashes and one Exclusive-OR operation (i.e., $2T_h + T_c$). On the other hand, Debiao He et al. [6] scheme consumes three hash operations, two inverse multiplications, one concatenation, one Exclusive-OR operation and two appending operations (i.e., $3T_h + 2T_m + T_c + T_e + 2T_a$).

The total computation cost of our authentication mechanism is $O(3T_h + 3T_a)$ and Debiao He et al scheme consumes $O(4T_{bp} + 14T_h)$. Jia-Lun Tsai et al. mutual authentication scheme consumes more computational overhead compare to the Debiao He et al. [6] scheme. We can conclude that our proposed scheme reduced computational overhead at the maximum extend by removing over usage of cryptographic operations.

Table 3. Running Time of Related Operations (in Millisecond)

<table>
<thead>
<tr>
<th>Related Operations</th>
<th>User Side</th>
<th>Servers Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bilinear Pairing ($T_{bp}$)</td>
<td>05.4277564</td>
<td>03.4545697</td>
</tr>
<tr>
<td>Scalar Multiplication ($T_a$)</td>
<td>02.1653048</td>
<td>01.1023660</td>
</tr>
<tr>
<td>One-way Hash operation ($T_h$)</td>
<td>00.3174836</td>
<td>00.1546821</td>
</tr>
<tr>
<td>Ex-OR operation ($T_c$)</td>
<td>00.0013785</td>
<td>00.0009859</td>
</tr>
<tr>
<td>Concatenation operation ($T_i$)</td>
<td>00.0002140</td>
<td>00.0000751</td>
</tr>
<tr>
<td>Inverse operation ($T_m$)</td>
<td>00.0032156</td>
<td>00.0001594</td>
</tr>
</tbody>
</table>

The registration and authentication phase running time of our scheme is represented in Figure 3 for different elliptic curves over prime fields. Here, we have considered 300 and 41442 records in registration and authentication phases respectively. The proposed scheme consumes less running time for Elliptic Curves Diffie-Hellman P521. The overall computation cost comparison of our scheme with Jia-Lun T et al.[5] and Debiao H et al. [6] is illustrated in Figure 3. The proposed scheme consumes less computation cost compare with Jia-Lun T et al. [5] and Debiao H et al. [6] schemes. The overall communication cost comparison of our scheme with Jia-Lun T et al.[5] and Debiao H et al. [6] is listed in Figure 4 for different elliptic curves over prime fields. The proposed scheme consumes less communication cost compare with Jia-Lun T et al. [5] and Debiao H et al. [6] schemes. Therefore, we can conclude that our proposed authentication scheme is computationally efficient and robust towards various
reply and impersonation attacks than the existing schemes.

CONCLUSION AND FUTURE DIRECTION

In this article, we proposed an efficient trustee-based authentication scheme for verifying genuineness of cloud communication entities. In the proposed scheme, identity provider computes session key materials and generates identity and access tokens for the user authentication. The existing scheme overall running time is 78.519 ms, communication cost is 3296 bits and performs four bilinear pairings and fourteen hash operations (i.e., $O(4T_{bp} + 14T_h)$). The proposed mechanism consumes minimum computational and communication costs than existing scheme i.e., the proposed scheme overall running time is 54.209748ms, communication cost is 2690 bits and performs three bilinear pairings and three hash operations (i.e., $O(3T_{bp} + 3T_h)$). Our investigation can be further extended to fog cloud computing in Internet of Things.

DECLARATIONS

Competing Interests

The authors Mr. Sabout Nagaraju and Dr.S.K.V. Jayakumar declare that they have no competing interests.
Authors’ Contributions

Mr. Sabour Nagaraju has made substantial contributions to conception, design, implementation, acquisition of test data, and performed experimental evaluation. Dr. S.K.V. Jayakumar has involved in revising it critically for important intellectual content, supervision of the research work and has given final approval of the version to be published. Both authors read and approved the final manuscript.

Acknowledgements

Thanks to the Pondicherry University administration for providing required hardware and software resources to carry out this work successfully.

REFERENCES


Authors' information

Sabout Nagaraju is currently working as assistant professor in Pondicherry University. He is graduated from G.Pulla Reddy Engineering College, Kurnool and did his post graduation at National Institute of Technology, Calicut. He is pursuing his Ph.D. from Pondicherry University, India. His professional experience spans over 11 years in various Engineering colleges and software industry. His areas of interest include Cloud Computing, Internet of Things and Big Data. He has published eight papers in international journals and seven papers in national/ international conferences.

Dr. S.K.V. Jayakumar is currently working as assistant professor in Pondicherry University. He obtained his B.E in Electrical and Electronics Engineering from Madurai Kamaraj University and did his M.E. in Computer Science and Engineering from Madras University. He has obtained his PhD from Pondicherry University. His teaching experience spans over 19 years and his research interest includes Web Services computing and cloud computing. He has published 29 research papers in International Journals and National and International Conferences.