SecAuthn: Provably Secure Multi-Factor Authentication for the Cloud Computing Systems

Sabout Nagaraju* and Latha Parthiban

Department of Computer Science, Pondicherry University Community College, Lawspet, Pondicherry - 605008, India; nagarajus.ucc@pondiuni.edu.in, lathaparthiban@yahoo.com

Abstract
Cloud computing is an advanced resource pooling framework which delivers an economical and more reliable Information and Communications Technology (ICT) solutions to the industry and academia. Cloud technology helps the stakeholders to avoid initial investment of costly infrastructure setup, licensing new software’s, training new personals and operational cost. Therefore, all size of IT organizations and individuals can make use of the cloud for boosting up their ICT needs. In parallel there has been backlash due to authentication access keys and credentials management issues in this new framework. As a result, in this article we have proposed a robust and privacy preserved Multi-Factor Authentication (MFA) scheme with efficient access keys distribution mechanism. The proposed MFA approach integrates the bio-metric fingerprint with user-id, password and One-Time Password (OTP) and upgrades the existing Single Sign-On (SSO) and two-factor authentications to multi-factor authentication. Our investigation not only provides the robust remote authentication in cloud but also preserves the privacy of the authentication credentials. The main shortcoming in our approach is that remote users and cloud service providers must trust the third party trustee. We have analyzed the completeness of our investigation using the GNY (Gong, Needham and Yahalom) logic. Finally, we reported the performance and robustness of our scheme with series of experiments.

Keywords: Authentication, Cloud Computing, Fingerprint, Privacy, Security, Trustee

1. Introduction
Information Technology is currently undergoing widespread transformation with adoption of cloud computing. In cloud environment, the enterprise and personal information access control aspects are managed out of the premises. Enterprise perimeter has disappeared and traditional methods to secure information assets have been eroded. Authentication solutions designed for on-premises will not work for the cloud computing systems. The sensitive data is shared more than ever through cloud services and via remote access as more and more employees use their mobile devices for work. This brings more risk of identity theft, data breaches and outages. A security challenge faced by enterprises is the security of passwords and more over multi-factor authentication is critical. To implement a secure multi-factor identification for all users, cloud services and mobile devices need to balance improved security with its costs and the potential burden on the IT department. Developing strong and convenient identity verification in a scalable cloud environment to accommodate growth is also a challenging task.

User identity verification is the fundamental operation to restrict the access to the sensitive data. Traditional authentications with the username and password/PIN are not enough to secure the cloud IT systems, because hackers are increasing with the appetite to score the next big information leak. In 1-4, authors described various tools and techniques to compromise the passwords. In 32, reported that Last Pass the CEO of Password Management Company says that web sites are regularly compromised with password files stolen, spear phishing attacks are up to 50% effective, 75% of people use the same password for multiple sites and password complexity rules help very little. So it is necessary to implement the high securable and reliable multi-factor authentication
scheme for the cloud systems without it becoming a burden for the IT department. Multi-factor authentication strongly protects against stolen passwords, where users must login with username, password and something else. The cost and security of the multi-factor authentication scheme used to access cloud applications and services depends on the selection of credentials parameters such as username, password and something else like secret question, soft tokens, bio-metrics, mobile-based one-time password, digital certificates, software tokens and smart cards.

Bio-metric–based authentications have huge advantages over the other parameter-based authentication schemes. Question-Answer based authentication approach can protect from stolen passwords, but it is not secure from the phishing attacks. Soft tokens can be generated using like Google authenticator App as a credential parameter for the authentication. It is very secure, impossible to guess or phish and easy to use once distributed, but it requires administrator to install app on mobile devices and works only on smart phones. Smart card-based authentication is more secure, but there is a chance to fail if password and card data are compromised. Sometimes smart card might be forged, stolen and it could be damaged. In contrast, bio-metric based authentication schemes have no such problems as well as provide high reliability, more convenience and robustness. Despite these advantages, fingerprint-based authentication has some challenges like availability of qualitative scanners with the remote devices, environment in which the user is capturing bio-metric data physically and verifying logically, fingerprint data cannot be changed or revoked and security and privacy of the fingerprint data. The above challenges and the problems motivated us to investigate a secure and privacy preserved fingerprint-based multi-factor authentication for the cloud computing systems.

1.1 Motivation

As part of the security in sensitive sectors like e-Governance, health care and finance, their online services need to be safeguard from inside and outside malicious use. The followings are the biggest and legitimate security and privacy concerns associated with the cloud-based platforms. These are the problems we have envisioned in our proposed research work.

- For some financial gain, dishonest cloud staff/cloud service provider may steal the authentication details of the remote users from the credentials database and they may use these details for acquiring user’s sensitive information.
- User’s bio-metric fingerprint details are unique and they may use these details for accessing more than one application. If the fingerprint details are compromised, then the users cannot change these details over the time.
- User’s personal information and activities can be tracked by using certain bio-metric fingerprint data.
- In cloud computing, performing authentication process on plain credentials is not securable because some authentication servers may be untrustworthy.
- Sometimes, an attacker may change the host IP/network address of the authorized user so that the request is coming from that altered system appears to be request coming from the legitimate user.
- Make sure that the user access keys such as master keys and one-time session keys are more secured, because these keys generates less cipher text and opponent can easily work on this cipher text. Access keys for the cloud-based environment that rented out from some cloud vendors need to appropriately managed and protected.
- Snooping user’s identities could be possible in cloud environment, where an attacker may eavesdrop on the credential communication channel and he/she may use replay attack.
- Similarly, dependency on geographic or legal jurisdiction that becomes another added point to consider, because certain laws in certain political jurisdictions may allow certain local agencies unrestricted access to the data that is hosted within their territory. For instance, the patriot law in the United States allows certain US agencies to demand access to the data which is stored in the US Union Territory. Enterprises are sensitive to this kind of a situation. Hence, need to take appropriate measures to ensure that authentication information still remains private regardless of whether it is stored in any territory.

From the organization’s perspective several risks are associated with cloud-based solutions. Some of the key risks we considered are summarized below:

- Complexity in compliance regulations and audit management.
- Dilution in functional, operational and technology control can lead to an impact on reputation, regulatory and business if service is hampered in cloud.
• Difficulties in sustaining security standards, regional privacy laws and information acts.
• Enterprise services will be locked in cloud and it is difficult to bring back in-house if required.
• Potentially cloud APIs are lacking in portability, so stakeholders cannot move from one cloud service provider to another.

1.2 Our Contribution
The following are the major contribution of our research:

1.2.1 Multi-factor Fingerprint bio-Metric Authentication (MFA)
The multi-factors are username and password, bio-metric fingerprint and OTP are used as key credentials in our authentication process. Where user ID and password shows what user know, bio-metric fingerprint represents what user is and OTP, master keys, session keys and nonce are used for verifying the users identity to servers and servers identities to the users. Our proposed trustee-based MFA provides a high-secure multi-stage identity verification process for validating the legitimacy of the end users.

1.2.2 Protection and Management of Access Keys
We used Station-to-Station Diffie-Hellman key exchange for preparing, securing and exchanging one-time session keys. Session keys are never stored in trustee/cloud server’s database due to privacy concerns. Nonces are used for handshaking and protecting alteration of requests in order to avoid untrustworthy servers and replay attacks.

1.2.3 Strong Privacy Preservation of User Credentials
In our proposed scheme, hashed credentials are just verified in the cloud authentication servers. Original key credentials are never revealed to cloud servers or trusted third party servers.

In our approach, hashed form of password and bio-metric fingerprint data will be at rest, transit and in use.

Advanced Encryption Standard (AES) algorithm is used for symmetric encryption/decryption of communication data between users and servers.

1.2.4 Provable Security and Privacy
With the above enhancements, our proposed authentication scheme provides a true protection for the user credentials in the cloud. Therefore the problems and risks envisioned in the previous section can be achieved.

This paper is further divided into six sections. Section 2 presents an overview of our proposed authentication scheme. Section 3 describes our proposed mechanism. The completeness of our multi-factor authentication protocol using GNY logic is described in Section 4. Section 5 reports the feasibility of proposed scheme. Literature reviews related to our research work are presented in Section 6. Section 7 summarizes the proposed work methodology.

2. Overview of our Proposed Scheme
In cloud computing environment, protecting IT stakeholder’s access credentials and encryption/decryption keys from the dishonest cloud staff and other malicious users is a challenging task. As part of this issue, we proposed an efficient privacy preserved multi-factor authentication scheme in cloud environment. The system level view of our proposed mechanism is depicted in Figure 1. In our scenario, clients will be registered with the Trusted Third Party Authenticator (TTPA) server and all the servers involved in the client communication need to register with each other and shares a secret key. The following are the key innovations of our proposed work:

• User can select their convenient User-Id (UID) and password (PWD) and the password must include at least one digit, one control character, uppercase and lowercase letters and one punctuation symbol which will be quite strong. We followed the proper rules and regulations to create, lockout and reset passwords as described in 23–26.
• Only the User-Id, phone numbers and primitive root \((g)\) and prime number \(p\) (which is \(g \mod p\)) are
kept in original form in the highly secured TTPA authentication database as shown in Table 1.

- Hashed Password (HPWD) and bio-metric fingerprint data (HBF) and encrypted form of secret random number (ERN) are also kept in the TTPA authentication database.
- User-Id will be verified in Trusted Third Party Authenticator Server (TTPAS), password is validated in the user module and bio-metric fingerprint will be verified in the Cloud Authentication Server (CAS). Finally the OTP will be verified in the Cloud Authorization Server (CARS).

A consumer who wants to avail a particular cloud online service needs to register with the enterprise, where customer has to submit his/her personal identification details such as permanent address proof, Mobile Number (MN), mail-id and most importantly Bio-Metric Fingerprint (BF). Enterprise does the user registration process with their trusted third party authenticator server. In this registration phase, user module takes UID, PWD, BF, MN and mail-id as input from the remote user and computes the Hashed Password (HPWD) using one-way hashing algorithm. Similarly user Bio-Metric Fingerprint template is also encoded and hashed (HBF) using cryptographically generated Random Number (RN) and SHA-2 family respectively. The cryptographically generated random number is also encrypted by using user’s fingerprint bio-metric data and that is indicated as ERN. The overview of the registration and authentication phases details are depicted in Figure 2. Once the registration is successful, then the UID, primitive root (g) value, modulo prime number (p) value and TTPA server public key details are sent to the user mail-id.

In authentication phase, user module takes User-Id (UID), password (PWD*), and Bio-Metric Fingerprint (BF*) as input from the remote user as shown in Figure 2 authentication phase. In this process, first the UID will be sent to the Trusted Third Party Authentication Server (TTPAS) for verification. TTPAS verifies the UID and its status; if it is valid then TTPAS retrieves the user HPWD, ERN, HBF from the Authentication Database (ADB) and sends to the user module, otherwise user will be rejected. Next, the user module prompts for user password to enter and then computes the Hashed Password h (PWD*) for user input password and then checks h (PWD*) with TTPAS HPWD, if both are same then it next prompts for user Bio-Metric Fingerprint to submit. User module then decrypts the ERN using user Bio-Metric Fingerprint and performs the encoding and hashing operations on user Bio-Metric Fingerprint and sends h (BF*) and TTPAS HBF to the cloud authentication server for verification. Cloud authentication server performs the matching, if both are same, then it allows the

Table 1. User’s credentials table

| Message (1) | Client Requests login access from the TTPA AS |
| PK<sub>TTPA</sub> | Trusted Third Party Authenticator public key |
| ID<sub>CS</sub> | User desired Service ID expecting from cloud |
| X<sub>A</sub> | Clients secrete value (i.e. X<sub>A</sub>=g<sup>a</sup> modulo p) for preparing one-time session key at TTPA AS |
| n<sub>1</sub> | Nonce to be used for handshaking between user and TTPA |
| UID | User conveys his/her identity to the TTPA AS |

| Message (2) | TTPA AS returns response to the client |
| SK | One-time session key to be used for TTPA AS and client encryption and decryption |
| K<sub>CAS</sub> | One-time session key to be used by client and CAS to communicate each other in a secure manner |
| n<sub>2</sub> | TTPA AS nonce (i.e. n<sub>2</sub>=n<sub>1</sub>+1) to be used for verifying clients and TTPA AS handshaking at client |
| HPWD | User’s TTPA database hashed password to be used for verifying user input password |
| ERN | User’s TTPA database encrypted form of random number to be used for encoding user input fingerprint |
| Token<sub>CAS</sub> | Token to be used by the user to access his/her desired service from CAS |
| K<sub>CAS</sub> | TTPA AS and CAS shared secrete key |
| NA<sub>C</sub> | Client’s network address to be used at CAS to verify present network address of the client |
| HFP | User’s TTPA database hashed bio-metric fingerprint to be used for verifying what user is |
| Y<sub>B</sub> | TTPA AS secrete value (i.e. Y<sub>B</sub>=g<sup>b</sup> modulo p) for preparing one-time session key at client |
user to access the enterprise online services from the cloud and it also provides OTP to the user for performing some important transactions. Otherwise user will be rejected.

Our proposed privacy preserved fingerprint-based authentication scheme is briefly illustrated in Figure 2.

Here, encoded and hashed Bio-Metric Fingerprint data of each user is verified in the cloud authentication server. To describe our authentication approach in Section 5, we introduce some important terminologies. We denote the registration phase Password as PWD, Hashed Password as HPWD, Bio-Metric Fingerprint data as BF, and encoded and Hashed Fingerprint Bio-Metric data as HBF. Further we use $\Delta$ as a matching algorithm for checking correctness of the hashed biometric data and the function $\delta_{RN}$ with Random Number RN is used for encoding fingerprint bio-metric data using exclusive OR operation. The function $\delta_{RN}$ cannot be computationally reversible without RN and will not affect on $\Delta$ matching results. The user module matches $h(PWD^*) = h(PWD)$ and the CAS verifies $\Delta(HBF, HBF^*) = (h(\delta_{RN}(BF)), h(\delta_{RN}(BF^*)))$. Thus, CAS cannot learn the original password and fingerprint bio-metric data, but still it can evaluate the correctness of the user legitimacy. Some other terminologies are given in Table 2.

### Table 2. Important terminologies used in messages used for verifying what user is

<table>
<thead>
<tr>
<th>Messages exchange between Client and CAS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Message (3)</strong> Client requests his/her desired service from the cloud</td>
<td></td>
</tr>
<tr>
<td>Token$_{CAS}$</td>
<td>Token to be used by CAS to verify the user bio-metric fingerprint to access his/her desired service from the cloud.</td>
</tr>
<tr>
<td>Legitimatiser$_C$</td>
<td>Prepared and sent by client to validate his/her bio-metric fingerprint legitimacy</td>
</tr>
<tr>
<td>$n_3$</td>
<td>Clients nonce (i.e. $n_3 = n_2 + 1$) to be used for verifying clients handshaking at CAS and it will be decremented and then checks with TTPA AS token nonce</td>
</tr>
<tr>
<td><strong>Message (4)</strong> CAS provides cloud service to the client</td>
<td></td>
</tr>
<tr>
<td>$K_{C,CARS}$</td>
<td>One-time session key to be used by client and CARS to communicate each other in a secure manner</td>
</tr>
<tr>
<td>$n_4$</td>
<td>CAS nonce (i.e. $n_4 = n_3 + 1$) to be used for verifying clients and CAS handshaking at client</td>
</tr>
<tr>
<td>OTP</td>
<td>One-time password to be used by user to performing some transactions with the sensitive cloud services</td>
</tr>
<tr>
<td>Token$_{CARS}$</td>
<td>OTP Token to be used by the user to perform his/her desired transaction on cloud sensitive services</td>
</tr>
<tr>
<td>$K_{CARS}$</td>
<td>CAS and CARS shared secret key</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Messages exchange between Client and CARS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Message (5)</strong> Client requests to perform his/her desired transaction</td>
<td></td>
</tr>
<tr>
<td>Token$_{CARS}$</td>
<td>Token to be used by CARS to verify the user OTP to perform his/her desired transaction with the cloud sensitive services.</td>
</tr>
<tr>
<td>Legitimatiser$_C$</td>
<td>Prepared and sent by client to validate his/her OTP legitimacy</td>
</tr>
<tr>
<td>$n_5$</td>
<td>Clients nonce (i.e. $n_5 = n_4 + 1$) to be used for verifying clients handshaking at CARS and it will be decremented and then checks with CAS token nonce</td>
</tr>
<tr>
<td><strong>Message (6)</strong> CARS returns mutual handshaking to the client</td>
<td></td>
</tr>
<tr>
<td>$n_6$</td>
<td>CARS nonce (i.e. $n_6 = n_5 + 1$) to be used for verifying clients and CARS handshaking at client</td>
</tr>
</tbody>
</table>
3. Our Multi-Factor Authentication Scheme

In this section we describe our privacy protected authentication approach in detail. In this approach, consumer registration and authentication will be performed using the following three phases. In our scheme, we assumed that the authentication and authorization servers involved in client communication need to be registered and share a secret key each other.

3.1 Initialization Phase

The TTPA Authentication Server (TTPA AS) chooses a larger prime value for \( p \), where \( p \) contains at least 300 digits and selects a primitive root value for \( g \), where \( g \) need not be larger. The \( p \) and \( g \) values will be used in login and authentication phase for preparing session keys between user and TTPA AS. TTPA also prepares the pair of private and public keys such as \( \text{PR}_{\text{TTPA}}, \text{PK}_{\text{TTPA}} \).

3.2 Registration Phase

In registration phase, consumer needs to register with the TTPA authentication server database as follows:

- A user \( U_i \) who wants to avail the enterprise cloud online services must produce a valid personal identity, mobile number and mail-id at the enterprise. In this process, the user needs to choose a User-Id and password where user’s selected password strength will be evaluated using the strong password checker and then need to pick a random number \( RN \) for Bio-Metric Fingerprint registration. Finally, the user’s fingerprint will be captured using high resolution scanner and a Bio-Metric Fingerprint template will be created. Thus the TTPA authentication server computes \( h (\text{PWD}) \), where \( h (.) \) is one-way hash function, \( HBF = h (\delta_{RN}(\text{BF})) = h ((\text{RN}\oplus\text{BF})) \) and \( E_{\text{BF}} (\text{RN}) = \text{ERN} \), where \( E_{\text{BF}}(.) \) is the encryption function using Bio-Metric Fingerprint BF as a key.10,31
- The TTPA authentication server stores UID, HPWD, HBF, ERN, MN, g, p and status in highly secured authentication database as shown in Table 1, where status denotes whether the registered UID is revoked or not. This credential table is kept in a highly secured authentication database.
- TTPA authentication server sends UID, g, p and TTPA public key to the user mail-id.

The login and authentication phase takes the following steps for validating correctness of the end user credentials as shown in the Figure 3.

- User Cloud Services ID (ID\(_C AS\)), \( X_A = g^a \mod p \) where user selects a secret integer \( a (a<p) \) and nonce \( n_1 \) where \( n_1 \) is a cryptographically generated pseudorandom number and these details will be encrypted using TTPA public key. Then the UID will be appended to the encrypted details and sends to the TTPA authentication server.
- TTPA authentication server obtains the UID from the client message and then checks it in the authentication database, if it found and valid, then TTPA AS chooses a secret integer \( b (b<p) \) and computes \( Y_B = g^b \mod p \). Next, TTPA AS computes a shared Secret Key (SK) as \( SK = X_B \mod p \) and then performs encryption on \( K_{C, CAS} \), HPWD, ERN using SK as \( E_{SK} [K_{C, CAS}] [n_2]\mod HPWD] [ERN] \) and then appends this result with the Token\(_{CAS}\) and \( Y_B \) as \( E_{SK} [K_{C, CAS}] [n_2]\mod HPWD] [ERN] || \text{Token}_{CAS} \) || \( Y_B \) and then sends to the user module.

Figure 3. Our proposed multi-factor authentication protocol
consider as n2. If UID is not found/invalid, then the login request will be rejected.

- User module computes the shared Secret Key (SK) as SK = Yb mod p using Yb obtained from the TTPA AS message and then obtains K_C,CAS = n2, HPWD, ERN by decrypting E_{SK} [K_{C,CAS}||n2||HPWD||ERN]. User module next checks the nonce value, if it is incremented by one then it finds the Hashed Password for user input password and then matches with the TTPA AS Hashed Password as h (PWD*) == h (PWD) if it is true, then decrypts RN as D_{BF*} (ERN) = RN using clients Bio-Metric Fingerprint data and then finds the encoded and hashes client Bio-Metric Fingerprint as h (δ_{RN} (BF*)) for user input Bio-Metric Fingerprint. User module then prepares the Legitimatise_c and appended it with the Token_{CAS} as Token_{CAS}||Legitimatise_c and sends to the CAS.

- CAS obtains the K_{C,CAS}, ID_{CAS}, NA_{C}, n2, HFP from Token_{CAS} using its Secret Key K_{CAS} and also retrieves the clients details ID_{CAS}, NA_{C}, n2, HFP from Legitimatise_c using K_{C,CAS} and then verifies the details received from TTPA AS with client details, if both are same then user is allowed to access the cloud services and also sends the OTP message such as E [K_{C,CARS}||n2||OTP] and Token_{CARS} to the client for only sensitive services.

- Whenever user need to perform transactions with sensitive services (e.g. net banking), then user module computes the Legitimatise_c and appends it with the Token_{CARS} and then sends to the Cloud Authorization Server (CARS).

- CARS decrypts the client message and verifies the OTP sent by CAS and client, if both are same then user is allowed to perform some transactions.

Table 2 describes the terminologies used in the client-servers messages of proposed authentication protocol.

Algorithm 1: Login and Authentication phase

1. TTPA if UID is found and valid then

   \[ C_r = (ID_{CAS}||X_a||n_1) \]
   \[ d(C_r) = (ID_{CS}||X_a||n_1) \]
   selects b (b<p)
   \[ Y_b = g^b mod p \]
   \[ SK = X_a^b mod p, n_2 = n_1 + 1 \]
   \[ C_2 = E_{SK}[K_{C,CAS}||n_2||HPWD||ERN] \]
   \[ Token_{CAS} = E [K_{C,CAS}||ID_{CAS}||NA_{C}||n_2||HFP] \]
   \[ m_2 = C_2||[Token_{CAS}||Y_b] \]

   TTPA TPU_r.

   If UID is not found or invalid, user request will be rejected

2. U, Inputs PWD*

   \[ SK = Y_b^* mod p \]
   \[ D_{SK}(C_r) = (K_{C,CAS}||n_2||HPWD||ERN) \]

   Checks n2 value, if it is incremented by 1, then

   Finds h (PWD*) and if h (PWD*) == HPWD, then

   Inputs BF*

   \[ A_{BF*} (ERN) = RN \]

   \[ Findsh(δ_{RN} (BF*)) = HFP \]
   and also computes

   \[ Legitimatise_c = E [ID_{CAS}||NA_{C}||n_2||HFP] \]
   \[ m_1 = Token_{CAS}||Legitimatise_c \]
   \[ U \rightarrow CAS \]

   CASD (Token_{CAS}) = (K_{C,CAS}||ID_{CAS}||NA_{C}||n_2||HFP)

   \[ D (Legitimatise_c) = (ID_{CAS}||NA_{C}||n_2||HFP) \]

   Checks TTPA AS token details with user Legitimatise details if ID_{CAS}, NA_{C}, nonce, HFP matches, then User is allowed to access cloud services and also prepares and sends OTP and Token_{CARS} details to the user as:

   \[ C_s = E [K_{C,CARS}||n_2||OTP] \]
   \[ Token_{CARS} = E [K_{C,CARS}||ID_{CAS}||NA_{C}||n_2||OTP] \]
3. $U$. User is allowed to access cloud services.

if user need to perform some transactions/ important actions on sensitive cloud services, then $D(C_i) = (K_{\text{CARS}}||n_i||\text{OTP})$ and checks nonce (n4) and then computes $\text{Legitimatis}_c = E(\text{ID}_{\text{CARS}}||\text{NA}_c||n_4||\text{OTP}^*)$

$m_5 = \text{Token}_{\text{CARS}}||\text{Legitimatis}_c$

$U \rightarrow \text{CARS}$

$\text{CARS}(\text{Token}_{\text{CARS}}) = (K_{\text{CARS}}||\text{ID}_{\text{CARS}}||\text{NA}_c||n_4||\text{OTP})$

$D(\text{Legitimatis}_c) = (\text{ID}_{\text{CARS}}||\text{NA}_c||n_5||\text{OTP}^*)$

Checks $\text{CARS}$ token details with $\text{user Legitimatis}$ details if $\text{ID}_{\text{CARS}}\text{NA}_c$ nonce, OTPmatches, then

User actions will be performed and also sends response to the user as $m_6 = E[n_6]$

$\text{CARS} \rightarrow U_i$

e() : A public-key encryption function’s with TTPA

$\text{AS}$ public key $PK_{\text{TTPA}}$

d() : A decryption function’s corresponding to e()

$A$: A random string extraction’s function.

$E_{k_{\text{cp}}}(\cdot)$ : A symmetric encryption’s function.

$D_{k_{\text{cp}}}(\cdot)$ : A symmetric decryption’s function corresponding $E_{k_{\text{cp}}}(\cdot)$

$\text{PWD}$* : The password which $U_i$ inputs.

$\text{BF}$* : The bio-metric fingerprint which $U_i$ submits.

$\text{OTP}$* : The one time password which $U_i$ inputs.

$C_i$: Cipher texts.

$m_i$: Messages.

4. Completeness of our Scheme

In this section we analyze the completeness of our proposed authentication protocol using belief logic. Burrows, Abadi and Needham (BAN) logic is the fundamental and popular belief logic which is widely used to analyse the completeness of various authentication schemes, but this logic has some shortcomings. Gong, Needham and Yalahom (GNY) logic is the extended version of the BAN logic. We used GNY logic to analyze our multi-factor authentication protocol. First, we describe important terminologies that we use in our belief logic and we re-describe our approach according to the GNY logic. Next, we describe our goals and finally we will report assumptions list.

4.1 Basic Terminologies and Statements

In this section we will define key terminologies which we used in our proposed GNY logic. Let $CP_i$ and $CP_j$ are the two credential parameters and we introduce the following rationale based on $CP_i$ and $CP_j$:

- $(CP_i, CP_j)$: Conjunction of two rationale $sCP_i$ and $CP_j$.
- $CP_i$: $CP$ is a credential parameter sent by user in login and authentication phase.
- $(CP_i)$: One way hashing function on $CP_i$.
- $(CP_i)_{\times_2}$ $(CP_i)_{\times_i}$: Asymmetric encryption and decryption of $CP_i$ using a public key $+k$ and a private key $-k$.
- $(CP_i)_{\times}$ $(CP_i)_{\times}$: Symmetric encryption and decryption of $CP_i$ using a key $k$.

In our proposed belief logic, the following are the statements which describes the properties of above rationale. Let $E_i$ and $E_j$ are the two entities which participate in the login and authentication approach.

- $E_i \equiv E_j$; $E_i$ is informed $E_j$.
- $E_i \equiv CP_i$; $E_i$ has a credential parameter $CP_i$.
- $E_i \equiv CP_i$; $E_i$ has conveyed $CP_i$.
- $E_i \equiv \#(CP_i)$; $E_i$ persuaded that $CP_i$ is generated from proper entity.
- $E_i \equiv \Phi(\text{CP}_i)$; $E_i$ feels that $CP_i$ is acceptable.
- $E_i \equiv E_j \leftrightarrow E_j$; $E_j$ persuaded that $E_i$ proper secret for $E_i$ and $E_j$.
- $E_i \Rightarrow E_j$; $E_j$ trusts that the $+k$ is proper public key for $E_i$.
- $E_i \Rightarrow E_j$; $E_j$ has authorization over $CP_i$.
- $E_i \equiv \times E_j$; $E_i$ informed to $E_j$ that he has not sent any messages in present session.

4.2 Protocol Transformation

Below we map our proposed authentication methodology into $E_i \Rightarrow E_j; CP_i$ form. We also convert some terminologies of our protocol to satisfy the GNY belief logic. In this approach, we also consider the TTPA AS public key as $+K$ and private key as $\sim K$. Here, the client is denoted as $C$, trusted third party authentication server is indicated as $S_1$, cloud authentication server is represented as $S_2$ and cloud authorization server is denoted as $S_3$.

- $C \rightarrow S_1$: $\{\text{ID}_{\text{CARS}}, X_A, x_i, k_i, \text{UID}\}$.
In the transformation $k_i$ to $k_j$ is considered as $SK$, $K_{CAP}$, $K_{CARS}$, $K_{CARS}$ in our actual protocol. Here, the client input PWD, BF and OTP we regard same as TTPA AS database details.

We then converted the protocol transformation into $E_i \mid CP_i$ and $E_i \not< E_j$ as given below. Here, if the rationale $CP_i$ and its terms are appears first time either in $E_i \mid CP_i$ or $E_i \not< E_j$ then those rationale and terms will be preceded with the star. Our authentication protocol transformation productions are described as follows:

$$S_i\Rightarrow C: \{K_3,n_2,HPWD,ERN\}_{K_1}\{K_3,ID_{CS},NA_c,n_2,HPFP\}_{K_2}, Y_b.$$ $C\rightarrow S_2: \{K_3,ID_{CS},NA_c,n_2,HPFP\}_{K_2}, \{ID_{CS},NA_c,n_2,HPFP\}_{K_3}\}
\rightarrow S_2\rightarrow C: \{K_3,n_2,OTP\}_{K_1}\{K_3,ID_{CS},NA_c,n_2,OTP\}_{K_4}
\rightarrow C\rightarrow S_3: \{K_3,ID_{CS},NA_c,n_2,OTP\}_{K_4}\{ID_{CS},NA_c,n_2,OTP\}_{K_5}\}
\rightarrow S_3\rightarrow C: \{n_3\}_{K_3}$

5. Goals

The goal of our proposed belief logic is categorized into four aspects as follows:

5.1 Message Content Authentication

In first flow, $S_1$ feels and believes that the client request is valid and recognizable.

$$S_1\Rightarrow \forall\{ID_{CS},X,A,n_1,UID\}.$$

In second flow, $C$ feels and believes that the $S_1$ response is valid and recognizable.

$$C\Rightarrow \forall\{ID_{CS},X,A,n_1,UID\}.$$

In third flow, $S_2$ feels and believes that the $C$ response is valid and recognizable.

$$S_2\Rightarrow \forall\{ID_{CS},X,A,n_1,UID\}.$$

5.2 Message Origin Authentication

In second flow, $C$ believes that $S_1$ originated the response;

$$C\equiv S_1: \{K_3,n_2,HPWD,ERN\}_{K_1}\{K_3,ID_{CS},NA_c,n_2,HPFP\}_{K_2}, Y_b.$$

In third flow, $S_2$ believes that $C$ conveyed the message;

$$S_2\equiv C: \{K_3,ID_{CS},NA_c,n_2,HPFP\}_{K_2}, \{ID_{CS},NA_c,n_2,HPFP\}_{K_3}\}.$$ In fourth flow, $C$ feels and believes that the $S_2$ response is valid and recognizable.

$$C\Rightarrow \forall\{K_3,n_2,OTP\}_{K_2}\{K_3,ID_{CS},NA_c,n_2,OTP\}_{K_3}\}.$$

In fifth flow, $S_3$ feels and believes that the client request is valid and recognizable.

$$S_3\Rightarrow \forall\{K_3,ID_{CS},NA_c,n_2,HPFP\}_{K_2}, \{ID_{CS},NA_c,n_2,HPFP\}_{K_3}\}.$$ In sixth flow, $C$ feels and believes that the $S_3$ response is valid and recognizable.

$$C\Rightarrow \forall\{n_3\}_{K_3}.$$ In fifth flow, $S_i$ believes that the client request is valid and recognizable.

$$S_i\Rightarrow \forall\{ID_{CS},X,A,n_1,UID\}.$$ In fourth flow, $C$ feels and believes that the $S_i$ response is valid and recognizable.

$$C\Rightarrow \forall\{K_3,n_2,OTP\}_{K_2}\{K_3,ID_{CS},NA_c,n_2,OTP\}_{K_3}\}.$$ In sixth flow, $C$ feels and believes that the $S_i$ response is valid and recognizable.

$$C\Rightarrow \forall\{n_3\}_{K_3}.$$
5.4 Generation of Session Keys

C and S₁ believes that K₁ is a one-time Secrete Key generated between C and S₁.

\[ C \equiv S₁ \leftrightarrow S₁. \]

S₁ and S₂ believes that K₂ is a Secrete Key shared between S₁ and S₂.

\[ S₁ \equiv S₂ \leftrightarrow S₂. \]

C and S₂ believes that K₃ is a temporal session Secrete Key shared between C and S₂.

\[ C \equiv S₂ \leftrightarrow S₂. \]

S₂ and S₃ believes that K₄ is a Secrete Key shared between S₂ and S₃.

\[ S₂ \equiv S₃ \leftrightarrow S₃. \]

C and S₃ believes that K₅ is a temporal session Secrete Key shared between C and S₃.

\[ C \equiv S₃ \leftrightarrow S₃. \]

5.5 Assumption List

To analyze the completeness of our proposed authentication protocol using belief logic we made the following list of assumptions:

- S₁ has public key +K, private key –K and a one-time session Secrete Key K₁.

\[ S₁ \equiv +K, S₁ \equiv –K, S₁ \equiv K₁ \]

S₁ is prepared one-time Secrete Key K₁ for encrypting session credential details. So that we assume S₁ believes K₁ is more securely prepared between S₁ and C.

\[ S₁ \equiv S₁ \leftrightarrow C. \]

- Since K₁ is first prepared by S₁ in our authentication approach, so that S₁ has K₁ and persuaded that K₁ is fresh and also assumes that K₁ will be prepared by C in the same way:

\[ S₁ \equiv S₁ \leftrightarrow \#(K₁). \]

- C is prepared one-time Secrete Key K₁ for decrypting session details. We assume that C believes K₁ is more securely prepared between C and S₁.

\[ C \equiv C₁ \leftrightarrow S₁. \]

- Since the one-time Secrete Key K₁ is prepared by, so that C has K₁ and trusts that K₁ is fresh

\[ C \equiv K₁, C \equiv \#(K₁). \]

- We assume that S₂ believes the Secrete Key K₂ is prepared by the authority S₁ and securely shared between S₂ and S₁.

\[ S₂ \equiv S₁ \leftrightarrow S₂. \]

- We assume that the client C believes the temporal session Secrete Key K₃ is prepared by the authority S₂ and securely shared between C and S₂.

\[ C \equiv S₂ \leftrightarrow S₂. \]

- We assume that S₁ believes the Secrete Key K₄ is prepared by the authority S₂ and securely shared between S₁ and S₂.

\[ S₁ \equiv S₂ \leftrightarrow S₂. \]

- We assume that the Shared Secrete key K₅ is prepared by the authority S₂ and securely shared between C and S₂.

\[ C \equiv S₂ \leftrightarrow C \leftrightarrow S₂. \]

6. Logic Analysis

By using GNY belief logic we analyzed our authentication protocol and we can also prove that our proposed methodology achieves its objectives. Below we described the logical postulates adoption of our proposed protocol to achieve its objectives, where we taken T₄ and T₃ logical postulates from the GNY logic²⁹.

6.1 The First Flow

\[ S₁ \equiv \{[ID_{CS}, X_{A}, n₁] X₁, UID]\}, S₂ \equiv K \]

\[ S₁ \equiv \{[ID_{CS}, X_{A}, n₁]\} \]

If the TTPA server S₁ is informed by the client C that the message \{ID_{CS}, X_{A}, n₁\} is encrypted with a public key
+K, then S₁ obtains \{ID_{CS}, X, n₁\} using corresponding private key -K. From the received and decrypted message:

\[ S₁ ≜ φ(UID), S₁ ≜ K \]

If S₁ believes that the client UID is recognizable and matches the private key -K, then S₁ accepts the client request and considers ID_{CS}, X, and n₁ for further authentication process. Therefore, we can understand that the TTPA server S₁ believes client request:

\[ S₁ ≜ φ\{ID_{CS}, X, n₁\}, S₁ ≜ +K \]

### 6.2 The Second Flow

If the TTPA server S₁ that the message \{K_{C, CAS}, n₂, HPWD, ERN\} is encrypted with a one-time session key SK which is generated at both the end, then C can obtain \{K_{C, CAS}, n₂, HPWD, ERN\} using SK. From the received message, the client’s decrypted contents:

\[ C ≜ φ\{n₂, HPWD, ERN\}, C ≜ SK \]

If the client module C feels that \( \text{Y}_b \) is recognizable and then C generates the one-time session key SK and entitled to believe that the rationale parameters n2, HPWD, ERN and \( \text{Y}_b \) are fresh. Therefore, C believes that the received credential parameters are fresh.

\[ C ≜ φ\{\text{Y}_b, \text{Y}_b, \text{Y}_b\}, C ≜ SK \]

If C believes that the decrypted and appended parameters are recognizable, then C also believes that the critical parameters such as nonce n₂, Hashed Password HPWD and ERN are fresh. Therefore, the client module strongly believes that the credentials generated and received in second flow are fresh.

\[ C \equiv \langle\{K_{C, CAS}, n₂, HPWD, ERN\}\rangle \]

\[ C \equiv \langle\{K_{C, CAS}, n₂, HPWD, ERN\}\rangle \]

The below given conditions are holds: 1) If C receives the rationale \{K_{C, CAS}, n₂, HPWD, ERN\} encrypted with SK; 2) C believes that all the credential components received are recognizable; 3) C generates SK; 4) C trusts that SK is fresh one-time secrete key for the second flow; 5) C entitled to trust that S₁ sent message is fresh. Therefore, the client module C validates the authentication received from the server S₁, if matched, and then client module believes that the client is legitimated entity.

If the client module C believes that the TTPA server S₁ sent n₂, HPWD, ERN, and \( \text{Y}_b \) are recognizable and matched, then client module accepts the client and considers RN and \( K_{C, CAS} \) for further identity validation at cloud authentication server. Therefore, we can understand that the client module trusts and continues the client and S₂ communications:

\[ C ≜ φ\{K_{C, CAS}, n₂, HPWD, ERN\} \]

According to the proposed belief logic, the client module C believes that the TTPA authentication server is honest. We assumes C|S₁| > S₁| >, and we form the following logical postulates for further adoption:

\[ C|S₁| > S₁| >C≡\langle\{K_{C, CAS}, n₂, HPWD, ERN\}\rangle \]

\[ C ≜ \langle\{K_{C, CAS}, n₂, HPWD, ERN\}\rangle \]

\[ C ≜ \langle\{K_{C, CAS}, n₂, HPWD, ERN\}\rangle \]

### 6.3 The Third Flow

If the TTPA server S₁ is informed by the client C that the message \{K_{C, CAS}, n₂, HPWD, ERN\} is encrypted with a TTPA AS and CAS shared Secrete Key \( K_{C, CAS} \), then S₂ obtains
\{K_{C,\text{CAS}},\text{ID}_\text{CS},\text{NA}_\text{C},n_2,\text{HFP}\} \text{ using corresponding Secrete Key } K_{\text{CAS}}. \text{ From the received and decrypted message.}
\begin{align*}
S_2 \&= & \Phi (K_{C,\text{CAS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_2,\text{HFP}) \wedge S_1 \wedge K_{CA}\text{S}
S_2 \&= & \Phi (K_{C,\text{CAS}},\text{ID}_\text{CS},\text{NA}_\text{C},n_2,\text{HFP})
\end{align*}

The below given conditions are holds: 1) If S\textsubscript{2} receives the rationale \{K_{C,\text{CAS}},\text{ID}_\text{CS},\text{NA}_\text{C},n_2,\text{HFP}\} encrypted with K\textsubscript{CAS}; 2) S\textsubscript{2} believes that all the rationale components received are recognizable; 3) S\textsubscript{2} decrypts rationale \{K_{C,\text{CAS}},\text{ID}_\text{CS},\text{NA}_\text{C},n_2,\text{HFP}\} using its shared Secrete Key K\textsubscript{CAS}; 4) S\textsubscript{2} trusts that K\textsubscript{CAS} is fresh one-time session key and used for decrypting client authentication details; 5) S\textsubscript{2} entitled to trust that C and S\textsubscript{1} sent message are fresh. The cloud authentication server S\textsubscript{2} validates the authentication details received from the server S\textsubscript{1} and C, if matched, then S\textsubscript{2} believes that the client has authenticated entity.

If S\textsubscript{2} believes that the client K\textsubscript{C,\text{CAS}},\text{ID}_\text{CS},\text{NA}_\text{C},n_2,\text{HFP} are recognizable, then K\textsubscript{C,\text{CAS}} will be used to decrypt the user details and then verifies these details with S\textsubscript{1} sent details. If ID\textsubscript{CS},\text{NA}_\text{C}, nonce and HFP are matched, then S\textsubscript{2} accepts the client as authenticated and allow to access the cloud services. Therefore, we can understand that the cloud authentication server S\textsubscript{2} believes TTPA server S\textsubscript{1} sent details for validating the user authentication details.

\begin{align*}
S_2 \&= & \Phi \{\text{ID}_\text{CS},\text{NA}_\text{C},n_2,\text{HFP}\}, C \equiv S_2 \equiv C \leftrightarrow S_2
S_2 \&= & \Phi \{\text{ID}_\text{CS},\text{NA}_\text{C},n_2,\text{HFP}\}
\end{align*}

6.4 The Fourth Flow

\begin{align*}
C \&= & \{\{\text{K}_{C,\text{CARS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}, \{\text{K}_{C,\text{CAS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}\}
S_2 \&= & \text{C\&S}_{C,\text{CARS}} \wedge S_1 \wedge \text{C\&S}_{C,\text{CAS}}
C \&= & \{\{\text{K}_{C,\text{CARS}}; n_4,\text{OTP}\}, \{\text{K}_{C,\text{CAS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}\}
\end{align*}

If the client C is informed by the cloud authentication server S\textsubscript{2} that the message \{K\textsubscript{C,\text{CARS}}; n_4,\text{OTP}\} is encrypted with the one-time session key K\textsubscript{CAS}, then C can obtain \{K\textsubscript{C,\text{CARS}}; n_4,\text{OTP}\} using K\textsubscript{CAS}. From the received message, the client's decrypted contents.

\begin{align*}
C \&= & \Phi (\text{K}_{C,\text{CARS}}; n_4,\text{OTP}) \wedge S_2 \wedge \text{C\&S}_{C,\text{CARS}}
C \&= & \Phi (\text{K}_{C,\text{CARS}}; n_4,\text{OTP})
\end{align*}

If the client module C feels that K\textsubscript{C,\text{CARS}}, n\textsubscript{4} and OTP are recognizable and then C checks the nonce, if it is valid, then entitled to believe that the rationale parameters K\textsubscript{C,\text{CARS}}, n\textsubscript{4} and OTP are fresh. Therefore, C believes that the received credential parameters are fresh.

\begin{align*}
C \&= & \Phi \{\{\text{K}_{C,\text{CARS}}; n_4,\text{OTP}\}, \{\text{K}_{C,\text{CAS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}\}
S_2 \&= & \text{C\&S}_{C,\text{CARS}} \wedge S_1 \wedge \text{C\&S}_{C,\text{CAS}}
C \&= & \Phi \{\{\text{K}_{C,\text{CARS}}; n_4,\text{OTP}\}, \{\text{K}_{C,\text{CAS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}\}
\end{align*}

The below given conditions are holds: 1) If C receives the rationale component \{K\textsubscript{C,\text{CARS}}; n_4,\text{OTP}\} encrypted with K\textsubscript{CAS} 2) C believes that all the rationale components received are recognizable; 3) C obtains \{K\textsubscript{C,\text{CARS}}; n_4,\text{OTP}\} using K\textsubscript{CAS}; 4) C trusts that K\textsubscript{C,\text{CAS}} is fresh temporal session key for the fourth flow; 5) C entitled to trust that S\textsubscript{2} sent message is fresh. Therefore, the client module C verifies the content of fourth flow message and believes that the received message components are recognizable and fresh.

If the client module believes that the cloud authentication server S\textsubscript{2} sent rationale components are recognizable and content are matched, then accepts the client and considers OTP and K\textsubscript{C,\text{CARS}} for further user authorization process at cloud authorization server. Therefore, we can understand that the client module trusts and continues the client and S\textsubscript{3} communications.

\begin{align*}
C \&= & \Phi \{\{\text{K}_{C,\text{CARS}}; n_4,\text{OTP}\}, \{\text{K}_{C,\text{CAS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}\}
S_2 \&= & \text{C\&S}_{C,\text{CARS}} \wedge S_3 \wedge \text{C\&S}_{C,\text{CAS}}
C \&= & \Phi \{\{\text{K}_{C,\text{CARS}}; n_4,\text{OTP}\}, \{\text{K}_{C,\text{CAS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}\}
\end{align*}

According to the proposed belief logic, the client module C believes that the cloud authentication server is honest. We assumes C[S\textsubscript{2}] \equiv S\textsubscript{3}, and we form the following logical postulates for further adoption.

\begin{align*}
C \&= & \{\{\text{K}_{C,\text{CARS}}; n_4,\text{OTP}\}, \{\text{K}_{C,\text{CAS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}\}
S_2 \&= & \{\{\text{K}_{C,\text{CARS}}; n_4,\text{OTP}\}, \{\text{K}_{C,\text{CAS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}\}
\end{align*}

6.5 The Fifth Flow

\begin{align*}
S_3 \&= & \{\{\text{K}_{C,\text{CARS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}, \{\text{K}_{C,\text{CAS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}\}
S_2 \&= & \text{C\&S}_{C,\text{CARS}} \wedge S_3 \wedge \text{C\&S}_{C,\text{CAS}}
S_2 \&= & \{\{\text{K}_{C,\text{CARS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}, \{\text{K}_{C,\text{CAS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\}\}
\end{align*}

If the cloud authorization server S\textsubscript{3} is informed by the client C that the message \{K\textsubscript{C,\text{CARS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\} is encrypted with a CARS and CAS shared secret key K\textsubscript{C,\text{CARS}} then S\textsubscript{3} obtains \{K\textsubscript{C,\text{CARS}}; \text{ID}_\text{CS},\text{NA}_\text{C},n_4,\text{OTP}\} using
corresponding Secret Key $K_{\text{CARS}}$. From the received and decrypted message,

$$S_2\equiv \Phi (K_{\text{CARS}}\text{ID}_{\text{CARS}} \text{NA}_{\text{C}}, n_2, \text{OTP}), S_2 \equiv K_{\text{CARS}}$$

$$S_3\equiv \Phi (K_{\text{CARS}}\text{ID}_{\text{CARS}} \text{NA}_{\text{C}}, n_3, \text{OTP})$$

The below given conditions are holds: 1) If $S_3$ receives the rationale $(K_{\text{CARS}}\text{ID}_{\text{CARS}} \text{NA}_{\text{C}}, n_3, \text{OTP})$ encrypted with $K_{\text{CARS}}$; 2) $S_3$ believes that all the rationale components received are recognizable; 3) $S_3$ decrypts rationale $(K_{\text{CARS}}\text{ID}_{\text{CARS}} \text{NA}_{\text{C}}, n_3, \text{OTP})$ using its shared Secret Key $K_{\text{CARS}}$; 4) $S_3$ trusts that $K_{\text{CARS}}$ is fresh one-time session key and used for decrypting client authorization details; 5) $S_3$ entitled to trust that C sent message is fresh. Therefore, the cloud authorization server $S_3$ verifies the authorization details received from the server $S_3$ and $C$, if matched, then believes that the client has authorized entity.

If $S_3$ believes that the client $\text{ID}_{\text{CARS}} \text{NA}_{\text{C}}, n_3$ and OTP are recognizable, then $K_{\text{CARS}}$ will be used to decrypt the user details and then verifies these details with $S_3$ sent details. If $\text{ID}_{\text{CARS}} \text{NA}_{\text{C}}, n_3$ and OTP are matched, then $S_3$ accepts the client as authorized and allow to perform sensitive actions on the cloud services. Therefore, we can understand that the cloud authorization server $S_3$ believes server $S_2$ details for validating user authorization details.

$$S_3\equiv \Phi (\text{ID}_{\text{CARS}} \text{NA}_{\text{C}}, n_3, \text{OTP}), C\equiv S_3 \leftrightarrow S_3$$

$$S_3\equiv \Phi (\text{ID}_{\text{CARS}} \text{NA}_{\text{C}}, n_3, \text{OTP})$$

6.6 The Sixth Flow

$$C\equiv \Phi \{n_6\} \ni S_3 \ni C\equiv \Phi \{n_6\}$$

If the client $C$ is informed by the cloud authorization server $S_3$ that the message $\{n_6\}$ is encrypted with the one-time session key $K_{\text{CARS}}$, then $C$ can obtain $\{n_6\}$ using $K_{\text{CARS}}$. From the received message, the client's decrypted contents.

$$C\equiv \Phi \{n_6\}, S_3 \ni C \ni K_{\text{CARS}}$$

$$C\equiv \Phi \{n_6\}$$

If the client module $C$ feels that $n_6$ is recognizable, and then $C$ checks the nonce, if it is valid, then entitled to believe that the rationale parameter $n_6$ is fresh. Therefore, $C$ believes that the received response is fresh.

$$C\equiv \Phi \{n_6\} \ni S_3 \ni C \ni K_{\text{CARS}}$$

$$C\equiv \Phi \{n_6\}$$

The below given conditions are holds: 1) If $C$ receives the rationale component $\{n_6\}$ encrypted with $K_{\text{CARS}}$; 2) $C$ believes that the received rationale is recognizable; 3) $C$ decrypts $\{n_6\}$ using $K_{\text{CARS}}$; 4) $C$ trusts that $K_{\text{CARS}}$ is fresh temporal session key for the six flow; 5) $C$ entitled to trust that $S_3$ sent message is fresh. Therefore, the client module $C$ verifies the content of message and believes that the received message is recognizable and fresh.

If the client module believes that the cloud authorization server $S_3$ sent rationale is recognizable and nonce is matched, then sensitive action performed by client is successful. Therefore, we can understand that the client module trusts and continues the client and $S_3$ communications.

$$C\equiv \Phi \{n_6\}, S_3 \ni C \ni K_{\text{CARS}}$$

$$C\equiv \Phi \{n_6\}$$

According to the proposed belief logic, the client module $C$ believes that the cloud authorization server is honest. We assumes $C\equiv S_3 \ni > S_3$, and we form the following logical postulates for further adoption.

$$C\equiv S_3 \ni > S_3, C\equiv \{n_6\}, C\equiv S_3 \ni > {n_6}, S_3 \ni C \ni S_3$$

$$C\equiv S_3 \ni C \ni S_3$$

7. Experimental Evaluation

The objective of this section is to report the robustness of our proposed authentication scheme. Before presenting the experimental performance evaluation, we explain the experimental setup including login and fingerprint databases we used. Later we describe the performance and properties of our authentication scheme in terms of security, time taken for login and authentication process, etc. With the extensive analysis and experiments we show that our proposed mechanism not only provides truly secure authentication, but also preserves the privacy of the credentials and access keys.
7.1 Experimental Setup and Inputs

7.1.1 Setup

We implemented our authentication framework in MATLAB R2013a. We use a machine running Windows 764-bits with 4GB RAM, 2.0GHz Intel Core i7 processor and a fingerprint reader.

7.1.2 Databases

We use four disjoint Fingerprint Databases (FDB’s) which are taken from the FVC2006 database\textsuperscript{12}. Where database images are captured using four different sensors with the cooperation of 150 heterogeneous participants includes industrial, academic and elderly people. The sensors used for capturing FVC2006 database fingerprint images details are given in Table 3. Each FDB contains 150 fingers and in-depth 12 samples per finger (i.e., $150 \times 12 = 1800$). Samples were of exaggerated distortion, dry/wet impressions and large amount of displacement and rotations. Each FDB is divided into two disjoint sub-databases as follows:

- FDB1-A, FDB2-A, FDB3-A and FDB4-A, where each sub-databases stores 140 fingerprint samples of their corresponding FDB.
- FDB1-B, FDB2-B, FDB3-B and FDB4-B, where each sub-database stores ten fingerprint samples of their corresponding FDB.

Where, B sub-database contains the most difficult fingerprint images, which can be used for evaluating detection strength of the authentication schemes. We generated 25000UID’s and PWD’s using GNU-licensed open source data generator tool\textsuperscript{13}.

1.1.3 Performance Evaluation

In our approach we used elliptic curve cryptosystem\textsuperscript{11} for public-key encryption/decryption and it takes only one modular multiplication. Also five symmetric encryption/decryptions, one exclusive-OR and one hash operation are required for each user authentication. Solutions\textsuperscript{19,21} requires minimum of two modular exponentiations for each user. In our protocol, a new idea is proposed where the user is allowed to select a User-Id (UID) and password, not decided by the cloud credential server, so that user can memorize easily. In\textsuperscript{18,22} mechanisms authentication servers decide UID’s and passwords for remote users. The solutions\textsuperscript{18,20} are the timestamp based, where the clock synchronization is required between the user and the server computers and the login message transmission delay time also limited. In our approach we used the nonce to eliminate the transmission and clock synchronization delay time and also avoids masquerade, eavesdrop and other replay attacks. In\textsuperscript{18,20,21} authors do not consider the phishing, Distributed Denial-of-Service (DDoS), man-in-the-browser and cross-site attacks. Our proposed authentication framework not only performs the credentials validation in CAS, but also provides the login and authentication credentials privacy. Mechanisms proposed in\textsuperscript{18,19,22} are not suitable for accessing sensitive online services in the cloud. Table 4 provides the performance comparisons of our approach with other mechanisms. To the best of our knowledge, our approach is an efficient multi-factor fingerprint bio-metric authentication

<table>
<thead>
<tr>
<th>Table 4. Performance comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.Jyoti Choudhury et al.\textsuperscript{18}</td>
</tr>
<tr>
<td>Ping Wang et al.\textsuperscript{19}</td>
</tr>
<tr>
<td>B.Rohitash Kumar et al.\textsuperscript{20}</td>
</tr>
<tr>
<td>Wenyi Liu et al.\textsuperscript{21}</td>
</tr>
<tr>
<td>Hong Liu et al.\textsuperscript{22}</td>
</tr>
<tr>
<td>Our Approach</td>
</tr>
</tbody>
</table>

C1: Requires low computation cost.
C2: The user is allowed to select a user-id (UID) and password, not decided by the cloud server.
C3: The clock synchronization is not required between the user and server computers.
C4: Robust towards phishing, Distributed Denial-of-Service (DDoS), man-in-the-browser and cross-site attacks.
C5: Not only performs the credentials validation in the CAS, but also provides the login and authentication credentials privacy.
C6: Suitable for accessing enterprise sensitive online services in the cloud.

Table 3. Details of sensors used in Fvc2006

<table>
<thead>
<tr>
<th>Database</th>
<th>Sensor Type</th>
<th>Resolution</th>
<th>Image Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDB1</td>
<td>Optical</td>
<td>569 dpi</td>
<td>400x560(224Kpixels)</td>
</tr>
<tr>
<td>FDB2</td>
<td>Electric Field</td>
<td>250 dpi</td>
<td>96x96(9Kpixels)</td>
</tr>
<tr>
<td>FDB3</td>
<td>Thermal sweeping</td>
<td>500 dpi</td>
<td>400x560(200Kpixels)</td>
</tr>
<tr>
<td>FDB4</td>
<td>SFInGe v3.0</td>
<td>500 dpi</td>
<td>288x384(108Kpixels)</td>
</tr>
</tbody>
</table>
approach which provides fingerprint bio-metric security and privacy in a cloud-based environment.

8. Results

We validated the correctness performance of our proposed fingerprint-based authentication protocol by using a series of experiments with the combination of 150 UID's and PWD's and four FVC2006 fingerprint databases. We set different time window bounds on FVC2006 databases for matching the correctness of given fingerprints in terms of False Negative Rate (FNR) and False Positive Rate (FPR). The False Negative Rate means the rate of genuine match or rejection of genuine claims and was calculated as $\frac{t_p}{(t_p + f_n)} \times 100\%$, where $f_n$ is the total number of false negative and $t_p$ is the total number of true positive. The false positive rate means the rate of impostor match or acceptance of impostor claims and was computed as $\frac{t_n}{(t_n + f_p)} \times 100\%$, where $t_n$ is considered as the total number of true negative and $f_p$ is taken as the total number of false positive.

The recognition performance of our proposed approach for FVC2006 databases is reported in Figure 4, where x-axis indicates databases DB1, DB2, DB3 and DB4 and y-axis indicates FNR and FPR percentages. We have set four different time window bounds such as 5, 10, 16 and 20 minutes for each database and we find out recognition rates. Our proposed approach substantially produced better fingerprint recognition rate than the existing fingerprint-based works$^{19,35,36}$. Figures 5 and 6 reports the False Negative and False Positive Rates comparison study of our scheme with other Bio-Metric Fingerprint-based schemes in cloud environment.

We find out the Rejection Enrollment (RE), Rejection Matching (RM), Average Enrollment Time (AET), Average Matching Time (AMT), Equal Error Rate (EER) and Revised EER (REER) over the FVC2006 databases as shown in Table 5. The EER, we consider as a unit of measure of fingerprint recognition performance and it denotes where the FNR and FPR are equal. The average EER of our mechanism for the FVC2006 databases is 1.44%. From the Table 4 we can understand that the EER little varies for each input fingerprint database of different

<table>
<thead>
<tr>
<th>UID</th>
<th>HPWD</th>
<th>ERN</th>
<th>HBF</th>
<th>MN</th>
<th>p</th>
<th>g</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>UID_1</td>
<td>HPWD_1</td>
<td>ERN_1</td>
<td>HBF_1</td>
<td>MN_1</td>
<td>p_1</td>
<td>g_1</td>
<td>Valid/Invalid</td>
</tr>
<tr>
<td>UID_2</td>
<td>HPWD_2</td>
<td>ERN_2</td>
<td>HBF_2</td>
<td>MN_2</td>
<td>p_2</td>
<td>g_2</td>
<td>Valid/Invalid</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>UID_i</td>
<td>HPWD_i</td>
<td>ERN_i</td>
<td>HBF_i</td>
<td>MN_i</td>
<td>p_i</td>
<td>g_i</td>
<td>Valid/Invalid</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Figure 5. Comparison of false negative rates.

Figure 6. Comparison of false positive rates.
sensor type. For example, the FDB4 has more equal error rate (i.e. 1.66%) when compared to FDB1 EER value (i.e., 1.15%) because these two databases differ in resolution and image sizes. Our scheme generated better equal error rate than the existing fingerprint-based works\textsuperscript{19,35,36} and comparison study is reported in Figure 7.

### 9. Related Work

Developing an efficient multi-factor authentication and key management approach for cloud-based platform is an open problem. Very few literatures are existing as a part of this problem in recent years. Our related work is divided into two parts; First we present the various traditional authentication mechanisms and next we report cloud-based authentication approaches.

Several traditional multi-factor authentication approaches have been designed to integrate the fingerprint bio-metrics with smart-card and/or password authentication. In\textsuperscript{5}, Lee et al. developed a User Identity Verification approach through smart cards; where the registered user supplies his/her fingerprint bio-metric samples and password in login process. In this scheme password tables are not required, but fingerprint and smart-card tables are required for validating the user’s identities. However, this mechanism was broken by the authors\textsuperscript{6,7}. In\textsuperscript{6} pointed out that Lee’s authentication approach cannot protect conspiring attack. Lin et al.\textsuperscript{7} discovered that an authorized user can make any number of fake valid credentials to masquerade other authorised users. Lin et al.\textsuperscript{7} discovered a scheme that maps the password and fingerprint into super password and enables authorized users to the password off-line. This approach cannot resist an impersonation attack\textsuperscript{15}. Yoon et al.\textsuperscript{15} presented a solution to resist this attack. This improved solution was broken by Lee et al.\textsuperscript{16} and they made further enhancement in this scheme. This solution is not broken till now, but it failing in checking some bio-metrics at server side. A MFA privacy preserving protocol has been proposed by Bhargav et al.\textsuperscript{17} using multi-factors namely password, a random string and a fingerprint. In this scheme they formed a cryptographic key by using multi-factors for identity verification. The problem with this scheme is in authentication phase each user needs to find expensive modular exponential computations. The above traditional multi-factor authentication mechanisms, however, do not suitable for cloud-based environment and the approaches of\textsuperscript{5–7,15,16} do not consider the privacy of the user credentials.

In recent years some cloud-based authentication mechanisms have been proposed for validating user credentials. A. J. Choudhury et al.\textsuperscript{18} presented an authentication framework to integrate the user ID and password with smartcard. This scheme is not enough strong for enterprises to protect intellectual properties, because it can easily compromise to replay and man-in-the-middle attacks. There are Bio-Metric Fingerprint-based works in cloud computing\textsuperscript{19,35,36}. In\textsuperscript{19}, Ping Wang et al. described a secret-splitting authentication method for enhancing cloud security using smart-card. In this scheme user id, password and one part of the encrypted bio-metric fingerprint data are stored in a smart-card and another part of the encrypted fingerprint template will be stored in the cloud database for user authentication. This approach preserves the credential and access keys privacy in the cloud, but it is not suitable for accessing cloud online services. Rohitash Kumar B et al.\textsuperscript{20} proposed a MFA framework using the OTP and IMEI number as authentication secrets. In\textsuperscript{21}, W. Liu et al. described a multi-factor cloud authentication approach using user password and secure user profile. However, the schemes\textsuperscript{20,21} reveal the user credentials to the cloud insiders and not suitable to achieve our problems, because here authors do not consider the privacy of the user credentials. Hong Liu et al.\textsuperscript{22} discovered a privacy-preserving authentication scheme based on the

### Table 5. Performance of our approach on the four Fvc2006 databases

<table>
<thead>
<tr>
<th>Database</th>
<th>EER</th>
<th>REER</th>
<th>RE</th>
<th>RM</th>
<th>AET</th>
<th>AMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDB1</td>
<td>1.15%</td>
<td>1.15%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.23 s</td>
<td>0.18 s</td>
</tr>
<tr>
<td>FDB2</td>
<td>1.49%</td>
<td>1.49%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.53 s</td>
<td>0.19 s</td>
</tr>
<tr>
<td>FDB3</td>
<td>1.48%</td>
<td>1.48%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.74 s</td>
<td>0.14 s</td>
</tr>
<tr>
<td>FDB4</td>
<td>1.66%</td>
<td>1.66%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.76 s</td>
<td>0.21 s</td>
</tr>
<tr>
<td>Avg.</td>
<td>1.44%</td>
<td>1.44%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.56 s</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7. Equal error rate comparison study.
shared authority details for data sharing. This theoretically proved approach helps for multi-user collaborative applications. To address our problems stated in Section 2, the user credentials and access keys should not be revealed to any cloud malicious insiders and outsiders. Our proposed fingerprint-based authentication scheme achieves the security and privacy concerns related to the remote user credentials and access keys in online cloud services.

10. Conclusion and Future Direction

Cloud computing is the present and futuristic resource pooling paradigm which converges with the Internet of Things (IoT). However, there are authentication and key management issues to be resolved. Identifying users is not an easy task in cloud. As a result in this article we proposed a provably secure multi-factors authentication scheme with trusted third party. In our approach, trustee distributes the authentication tokens on behalf of cloud service providers and allows the cloud servers just to verify the hashed key credential data. This approach also ensures the mutual authentication of the communication entities. We used multi-party station to station Diffie-Hellman key exchange protocol which overcomes many key management problems. Our proposed mechanism preserves the privacy of the remote authentication details in the cloud and significantly helps to protect the stakeholder’s sensitive information from the inside and outside malicious attackers. Our work and many existing cloud-based authentication works are still centralized and are yet to be transformed to a distributed or collaborative cloud paradigm.

11. Competing Interests

The authors Mr. Sabout Nagaraju and Dr. Latha Parthiban declared that they have no competing interests.

Has published research papers in 26 international journals and presented papers in 22 international and national conferences. She has also published a book in the area of computer aided diagnosis.

12. Acknowledgements

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

13. References