Indian Journal of Science and Technology, *Vol 9(9), DOI; 10.17485/ijst/2016/v9i9/81070, March 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

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

**Author for correspondence*

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 secrete 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[10,14,15,34]. 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 phone's. 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[8,9]. 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[33].

- 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 API's 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-MetricAuthentication (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.

Nonce's 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 severs.

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 secrete 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 modulo p*)* are

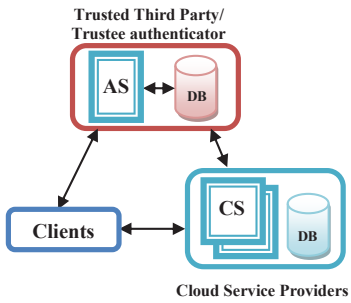

Figure 1. System level view of our proposed. mechanism

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 secrete 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

Messages exchange between Client and TTPA AS	
Message (1)Client Requests login accessfrom the TTPA AS	
PK_{TTPA}	Trusted Third Party Authenticator public key
ID_{CS}	User desired Service ID expecting from cloud
X_A	Clients secrete value (i.e. $X_A=g^a$ modulo *p)* for preparing one-time session key at TTPA AS
n_1	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
$K_{C,CAS}$	One-time session key to be used by client and CAS to communicate each other in a secure manner
n_2	TTPA AS nonce (i.e. $n_2=n_1+1$) 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
$Token_{CAS}$	Token to be used by the user to access his/her desired service from CAS
K_{CAS}	TTPA AS and CAS shared secrete key
NA_C	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_B	TTPA AS secrete value (i.e. $Y_B=g^b$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.

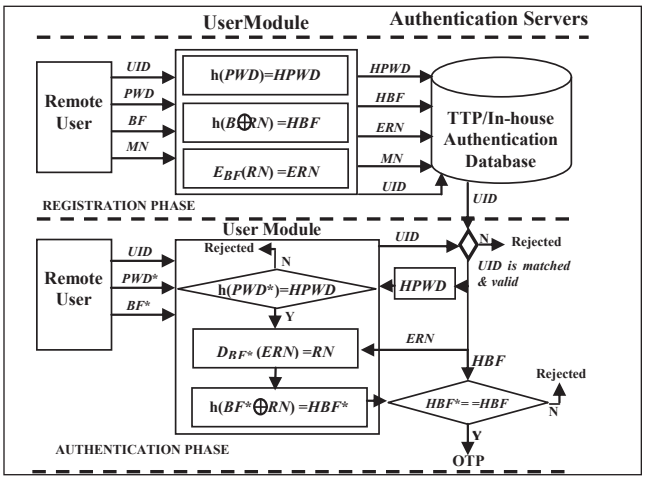

Figure 2. Block diagram of our proposed. authentication scheme

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 Bio-Metric Fingerprint data as HBF. We indicate the user authentication phase Password as PWD*, Hashed Password as HPWD*, Fingerprint Bio-Metric data as BF* and encoded and Hashed Fingerprint Bio-Metric data as HBF*. Further we use Δ as a matching algorithm for checking correctness of the hashed bio-metric data and the function δ_{RN} with Random Number RN is used for encoding fingerprint bio-metric data using exclusive *OR* operation. The function δ_{RN} cannot be computationally reversible without RN and will not affect on Δ matching results. The user module matches *h* (PWD*)==*h* (PWD)and the CAS verifies Δ (HBF, HBF*) = (h(δ_{RN}(BF)), h(δ_{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

Messages exchange between Client and CAS	
Message (3)Client requests his/her desired service from the cloud	
$Token_{CAS}$	Token to be used by CAS to verify the user bio-metric fingerprint to access his/her desired service from the cloud.
$Legitimatise_C$	Prepared and sent by client to validate his/her bio-metric fingerprint legitimacy
n_3	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
Message (4)CAS provides cloud service to the client	
$K_{C, CARS}$	One-time session key to be used by client and CARS to communicate each other in a secure manner
n_4	CAS nonce (i.e. $n_4=n_3+1$) to be used for verifying clients and CAS handshaking at client
OTP	One-time password to be used by user to performing some transactions with the sensitive cloud services
$Token_{CARS}$	OTP Token to be used by the user to perform his/her desired transaction on cloud sensitive services
K_{CARS}	CAS and CARS shared secrete key
Messages exchange between Client and CARS	
Message (5)Client requests to perform his/her desired transaction	
$Token_{CARS}$	Token to be used by CARS to verify the user OTP to perform his/her desired transaction with the cloud sensitive services.
$Legitimatise_C$	Prepared and sent by client to validate his/her OTP legitimacy
n_5	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
Message (6)CARS returns mutual handshaking to the client	
n_6	CARS nonce (i.e. $n_6=n_5+1$) to be used for verifying clients and CARS handshaking at client

3. Our Multi-FactorAuthentication 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 secrete 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 PR_{TTPA}, PK_{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 (PWD), here h (.) +indicated as one-way hash function, HBF = h (δ_{RN} (BF)) = h ((RN⊕BF)) and E_{BF*} (RN) =ERN, where E_{BF} (.) is the encryption function using Bio-Metric Fingerprint BF∗ as a key[30,31].
- 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 unrevoked 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_{CS}), $X_A = g^a$ modulo *p* where user selects a secrete 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 secrete integer *b* (*b*<p) and computes $Y_B = g^b$ mod *p*. Next, TTPA AS computes a shared Secrete Key (SK) as SK = X_A^b modulo p and then performs encryption on $K_{C, CAS,}$ n_2, HPWD, ERNusing SK as E_{SK} [$K_{C, CAS}$||n_2||HPWD||ERN] and then appends this result with the $Token_{CAS}$ and Y_B as E_{SK} [$K_{C\,CAS}$||n_2||HPWD||ERN]||$Token_{CAS}$||Y_B and then sends to the user module. Where, TTPA AS increments the nonce by one that is

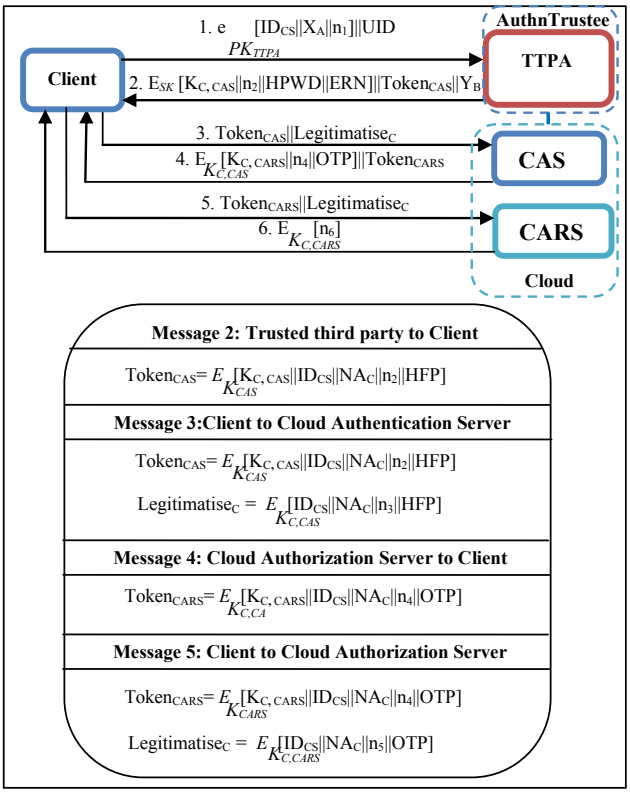

Figure 3. Our proposed multi-factor authentication protocol

consider as n_2. If UID is not found/invalid, then the login request will be rejected.

- User module computes the shared Secrete Key (SK) as $SK = Y_B^{\ a}$ modulo p using Y_B obtained from the TTPA AS message and then obtains $K_{C, CAS}$, n_2, HPWD, ERN by decrypting E_{SK} $[K_{C, CAS}||n_2||HPWD||ERN]$. 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 $h(\delta_{RN}(BF*)$ 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_{CS}, NA_C, n_2, HFP from $Token_{CAS}$ using its Secrete Key K_{CAS} and also retrieves the clients details ID_{CS}, NA_C, n_3, 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 $[K_{C, CARS}||n_4||OTP]$ 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

U_i Inputs UID and selects *a (a<p)* and n_1

$$X_A = g^a \bmod p$$

$$C_1 = e\ [ID_{CS}||X_A||\ n_1]$$

$$m_1 = C_1||UID$$

$$U_i \rightarrow TTPA$$

1. TTPA if UID is found and valid then

$$d\ (C_1) = (ID_{CS}||X_A||\ n_1)$$

selects *b (b<p)*

$$Y_B = g^b \bmod p$$

$$SK = X_A^{\ b} \bmod p,\ n_2 = n_1+1$$

$$C_2 = E_{SK}[K_{C, CAS}||n_2||HPWD||ERN]$$

$$Token_{CAS} = E\ [K_{C, CAS}||ID_{CS}||NA_C||n_2||HFP]$$

$$m_2 = C_2||Token_{CAS}||Y_B$$

$$TTPA\ TPU_i,$$

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

2. U_i Inputs PWD*

$$SK = Y_B^{\ a} \bmod p$$

$$D_{SK}(C_2) = (K_{C, CAS}||n_2||HPWD||ERN)$$

Checks n_2 value, if it is *incremented* by 1, then

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

Inputs *BF**

$$A_{BF*}\ (ERN) = RN$$

$$Findsh(\delta_{RN}\ (BF*) = HFP$$ and also computes

$$Legitimatise_C = E\ [ID_{CS}||NA_C||n_3||HFP]$$

$$m_3 = Token_{CAS}||Legitimatise_C$$

$$U_i \rightarrow CAS$$

$$CASD\ (Token_{CAS}) = (K_{C, CAS}||ID_{CS}||NA_C||n_2||HFP)$$

$$D\ (Legitimatise_C) = (ID_{CS}||NA_C||n_3||HFP)$$

Checks *TTPA AS token* details with *user Legitimatise* details *if* ID_{CS}, 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:

$$C_3 = E\ [K_{C, CARS}||n_4||OTP]$$

$$Token_{CARS} = E\ [K_{C, CARS}||ID_{CS}||NA_C||n_4||OTP]$$

$$m_4 = C_3 || Token_{CARS}$$

$$_{CAS} \rightarrow U_i,$$

3. U_i User is allowed to access cloud services.

if user need to perform some transactions/ important actions on sensitive cloud services, then $D(C_3)=(K_{C, CARS}||n_4||OTP)$ and checks *nonce* (n4) and then computes $Legitimatise_C = E[ID_{CS}||NA_C||n_5||OTP*]$

$$m_5 = Token_{CARS}||Legitimatise_{C.}$$

$$U_i \rightarrow CARS$$

$$CARS\ D(Token_{CARS}) = (K_{C, CARS}||ID_{CS}||NA_C||n_4||OTP)$$

$$D(Legitimatise_C) = (ID_{CS}||NA_C||n_5||OTP*)$$

Checks *CAS token* details with *user Legitimatise* details *if* ID_{CS}, NA_C, *nonce*, *OTP*matches, *then*

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

$$CARS \rightarrow U_i$$

$e(\cdot)$: A public-key encryption function's with TTPA AS public key $PK_{TTPA,}$

$d(\cdot)$: A decryption function's corresponding to $e(\cdot)$

A: A random string extraction's function.

$E_{K(.)}(\cdot)$: A symmetric encryption'sfunction.

$D_{K(.)}(\cdot)$: A symmetric decryption'sfunction corresponding $E_{K(.)}(\cdot)$.

$PWD*$: The password which U_i inputs.

$BF*$: The bio-metric fingerprint which U_i submits.

$OTP*$: 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[27] 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[28]. Gong, Needham and Yahalom (GNY) logic[29] is the extended version of the BAN logic. We used GNY logic[29] 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_i is a credential parameter sent by user in login and authentication phase.
- (CP_i): One way hashing function on CP_i.
- $\{CP_i\}_{+k}$, $\{CP_i\}_{-k}$: Asymmetric encryption and decryption of CP_i using a public key $+k$ and a private key $-k$.
- $\{CP_i\}_k$, $\{CP_i\}^{-1}_k$: 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.

- $E_i \lhd E_j$: E_i is informed $E_{j..}$
- $E_i \ni CP_i$: E_i has a credential parameter $CP_{i.}$
- $E_i \sim CP_i$: E_i conveyed $CP_{i.}$
- $E_i \equiv \#(CP_i)$: E_i persuaded that CP_i is generated from proper entity.
- $E_i \equiv \Phi(CP_i)$: E_if eels that CP_i is acceptable.
- $E_i \equiv E_i \leftrightarrow E_j$: E_i persuaded that Sisa proper secrete for E_i and $E_{j.}$
- $E_i \Rightarrow E_j$: E_i trusts that's $+K$ is a proper public key for $E_{j.}$
- $E_i \Rightarrow CP_i$: E_i has authorization over $CP_{i.}$
- $E_i \lhd *E_j$: 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 $-K$. 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.

- $C \rightarrow S_1$: $\{\{ID_{CS}, X_A, n_1\}_{+K}, UID\}$.

- $S_1 \rightarrow C$: $\{\{K_3,n_2,HPWD,ERN\}_{K1},\{K_3,ID_{CS},NA_C,n_2,HFP\}_{K2},Y_B\}$
- $C \rightarrow S_2$: $\{\{K_3,ID_{CS},NA_C,n_2,HFP\}_{K2},\{ID_{CS},NA_C,n_3,HFP\}_{K3}\}$
- $S_2 \rightarrow C$: $\{\{K_5,n_4,OTP\}_{K3},\{K_5,ID_{CS},NA_C,n_4,OTP\}_{K4}\}$
- $C \rightarrow S_3$: $\{\{K_5,ID_{CS},NA_C,n_4,OTP\}_{K4},\{ID_{CS},NA_C,n_5,OTP\}_{K5}\}$
- $S_3 \rightarrow C$: $\{n_6\}_{K5}$

In the above transformation $_{K1}$ to K_5 is considered as SK, K_{CAS}, $K_{C,CAS}$, K_{CARS}, $K_{C,CARS}$ 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 $E_i \lhd E_j$ as given below. Here, if the rationale CP_i and its terms are appears first time either in $E_i \mid CP_i$ or $E_i \lhd E_j$ then those rationale and terms will be preceded with the star. Our authentication protocol transformation productions are described as follows:

- $S_1 \lhd \{*\{*ID_{CS}, *X_A, *n_1\}_{+K}, *UID\}_{\sim>} C \mid\equiv C \leftrightarrow S_1$.
- $C \lhd \{*\{*K_3, *n_2, *HPWD, *ERN\}_{K1}, *\{K_3,ID_{CS},*NA_C,*n_2,*HFP\}_{K2},*Y_B\}\}_{\sim>} S_1 \mid\equiv S_1 \leftrightarrow C$.
- $S_2 \lhd \{\{K_3,ID_{CS},NA_C,n_2,HFP\}_{K2},*\{ID_{CS},NA_C,*n_3,HFP\}_{K3}\}_{\sim>} C \mid\equiv C \leftrightarrow S_2$.
- $C \lhd \{*\{*K_5,*n_4,*OTP\}_{K3},*\{*K_5,ID_{CS},NA_C,n_4,OTP\}_{K4}\}\}_{\sim>} S_2 \mid\equiv S_2 \leftrightarrow C$.
- $S_3 \lhd \{\{K_5,ID_{CS},NA_C,n_4,OTP\}_{K4},*\{ID_{CS},NA_C,*n_5,OTP\}_{K5}\}_{\sim>} S_3 \mid\equiv S_3 \leftrightarrow C$.
- $C \lhd \{*n_6\}_{K5 \sim>} C \mid\equiv C \leftrightarrow S_3$.

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.

$$S_1 \mid\equiv \Phi\{\{ID_{CS}, X_A, n_1\}_{+K}, UID\}.$$

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

$$C \mid\equiv \Phi \{\{K_3,n_2,HPWD,ERN\}_{K1},\{K_3,ID_{CS},NA_C,n_2,HFP\}_{K2},Y_B\}.$$

In third flow, S_2 feels and believes that the client request is valid and recognizable.

$$S_2 \mid\equiv \Phi\{\{K_3,ID_{CS},NA_C,n_2,HFP\}_{K2},\{ID_{CS},NA_C,n_3,HFP\}_{K3}\}.$$

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

$$C \mid\equiv \Phi \{\{K_5,n_4,OTP\}_{K3},\{K_5,ID_{CS},NA_C,n_4,OTP\}_{K4}\}.$$

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

$$S_3 \mid\equiv \Phi \{\{K_5,ID_{CS},NA_C,n_4,OTP\}_{K4},\{ID_{CS},NA_C,n_5,OTP\}_{K5}\}.$$

In sixth flow, C feels and believes that the S_3 response is valid and recognizable.

$$C \mid\equiv \Phi \{n_6\}_{K5}.$$

5.2 Message Origin Authentication

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

$$C \equiv S_1 \mid\sim\{\{K_3,n_2,HPWD,ERN\}_{K1},\{K_3,ID_{CS},NA_C,n_2,HFP\}_{K2},Y_B\}.$$

In third flow, S_2 believest hat C conveyed the message;

$$S_2 \equiv C \mid\sim\{\{K_3,ID_{CS},NA_C,n_2,HFP\}_{K2},\{ID_{CS},NA_C,n_3,HFP\}_{K3}\}.$$

In fourth flow, C believes that S_2 sent the response;

$$C \equiv S_2 \mid\sim\{\{K_5,n_4,OTP\}_{K3},\{K_5,ID_{CS},NA_C,n_4,OTP\}_{K4}\};$$

In fifth flow, S_3 believes that C conveyed the message;

$$S_3 \equiv C \mid\sim\{\{K_5,ID_{CS},NA_C,n_4,OTP\}_{K4},\{ID_{CS},NA_C,n_5,OTP\}_{K5}\}.$$

In sixth flow, C believes that S_3 sent the response;

$$C \equiv S_3 \mid\sim\{n_6\}_{K5}.$$

5.3 Credentials Verification and Validation

In third flow, S_2 believes that C's Hashed Password was verified and also he/she has encoded and hashed Bio-Metric Fingerprint data for verification;

$$S_2 \mid\equiv C\ni\{h(\delta_{RN}(BF*)\}.$$

In fifth flow, S_3 believes that C's Hashed Password and Bio-Metric Fingerprint data was verified and also he/she has OTP for verification;

$$S_3 \mid\equiv C\ni\{OTP\}.$$

5.4 Generation of Session Keys

C and S_1 believes that K_1 is a one-time Secrete Key generated between C and S_1.

$$C|\equiv S_1 \equiv C \leftrightarrow S_1.$$

S_1 and S_2 believes that K_2 is a Secrete Key shared between S_1 and S_2.

$$S_1|\equiv S_2 \equiv S_1 \leftrightarrow S_2.$$

C and S_2 believes that K_3 is a temporal session Secrete Key shared between C and S_2.

$$C|\equiv S_2 \equiv C \leftrightarrow S_2.$$

S_2 and S_3 believes that K_4 is a Secrete Key shared between S_2 and S_3.

$$S_2|\equiv S_3 \equiv S_2 \leftrightarrow S_3.$$

C and S_3 believes that K_5 is a temporal session Secrete Key shared between C and S_3.

$$C|\equiv S_3 \equiv C \leftrightarrow S_3.$$

5.5 Assumption List

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

- S_1 has public key +K, private key –K and a one-time session Secrete Key K_1

$$S_1 \ni +K, S_1 \ni -K, S_1 \ni K_1$$

S_1 is prepared one-time Secrete Key K_1 for encrypting session credential details. So that we assume S_1believes K_1 is more securely prepared between S_1 and C.

$$S_1|\equiv S_1 \leftrightarrow C.$$

- Since $K_1$is first prepared by S_1 in our authentication approach, so that S_1 has K_1 and persuaded that K_1 is fresh and also assumes that K_1 will be prepared by C in the same way.

$$S_1 \ni K_1, S_1 \equiv \#(K_1).$$

- C is prepared one-time Secrete Key K_1 for decrypting session details. We assume that C believes K_1 is more securely prepared between C and S_1.

$$C|\equiv C_1 \leftrightarrow S_1.$$

- Since the one-time Secrete Key K_1 is prepared by, so that C has K_1 and trusts that K_1 is fresh

$$C \ni K_1, C \equiv \#(K_1).$$

- We assume that S_2 believes the Secrete Key K_2 is prepared by the authority S_1 and securely shared between S_1 and S_2.

$$S_2 \equiv S_1|\Rightarrow S_1 \leftrightarrow S_2.$$

- We assume that the client C believes the temporal session Secrete Key K_3 is prepared by the authority S_1 and securely shared between C and S_2.

$$C \equiv S_1|\Rightarrow C \leftrightarrow S_2.$$

- We assume that S_3 believes the Secrete Key K_4 is prepared by the authority S_2 and securely shared between S_2 and S_3.

$$S_3 \equiv S_2|\Rightarrow S_2 \leftrightarrow S_3.$$

- We assume that the Shared Secrete key K_5 is prepared by the authority S_2 and securely shared between C and S_3.

$$C \equiv S_2|\Rightarrow C \leftrightarrow S_3.$$

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 T_4 and T_3 logical postulates from the GNY logic[29].

6.1 The First Flow

$$S_1 \lhd \{\{ID_{CS}, X_A, n_1\}_{+K}, UID\}, S_1 \ni -K$$

$$S_1 \lhd \{ID_{CS}, X_A, n_1\}$$

If the TTPA server S_1 is informed by the client C that the message $\{ID_{CS}, X_A, n_1\}_{+K}$ is encrypted with a public key

$+K$, then S_1 obtains $\{ID_{CS}, X_A, n_1\}$ using corresponding private key $-K$. From the received and decrypted message:

$$S_1|\equiv \Phi\ (UID), S_1 \ni -K$$

$$S_1|\equiv \Phi\ \{\{ID_{CS}, X_A, n_1\}_{+K}\}$$

If S_1 believes that the client UID is recognizable and matches the private key –K, then S_1 accepts the client request and considers ID_{CS}, X_A and n_1 for further authentication process. Therefore, we can understand that the TTPA server S_1 believes client request:

$$S_1|\equiv \Phi\ \{\{ID_{CS}, X_A, n_1\}_{+K}, S_1 \ni +K$$

$$S_1|\equiv \Phi\ \{\{ID_{CS}, X_A, n_1\}_{+K}, UID\}$$

6.2 The Second Flow

$C \lhd \{\{K_{C,CAS}, n_2, HPWD, ERN\}_{SK}, \{K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP\}, Y_B\}\ S_1, C \ni SK,\ C \equiv S_1|\equiv \#(SK),\ S_2 \ni K_{CAS}$

$C \lhd \{\{K_{C,CAS}, n_2, HPWD, ERN\}_{SK}, \{K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP\}, Y_B\}$

If the client C is informed by the TTPA server S_1 that the message $\{K_{C,CAS}, n_2, HPWD, ERN\}SK$ is encrypted with the one-time session key *SK* which is generated at both the end, then C can obtain $\{K_{C,CAS}, n_2, HPWD, ERN\}$ using *SK*. From the received message, the client's decrypted contents:

$$C|\equiv \Phi\ (n_2, HPWD, ERN), C \ni SK$$

$$C|\equiv \Phi\ \{\{K_{C,CAS}, n_2, HPWD, ERN\}_{SK}\}$$

If the client module C feels that 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 Y_B are fresh. Therefore, C believes that the received credential parameters are fresh.

$$C|\equiv \Phi\ (\{n_2, HPWD, ERN\}_{SK}, Y_{B),C} \ni SK, C \equiv \#(SK)$$

$$C|(S\{\{K_{C,CAS}, n_2, HPWD, ERN\}_{SK}, Y_B\}$$

If C believes that the decrypted and appended parameters are recognizable, then C also believes that the critical parameters such as nonce n_2, Hashed Password HPWD and ERN are fresh. Therefore, the client module strongly believes that the credentials generated and received in second flow are fresh.

$C \lhd_* (\{K_{C,CAS}, n_2, HPWD, ERN\}_{SK} || Y_B), C \ni SK, C|\equiv S_1 \equiv C \leftrightarrow S_1,\ C|\equiv \Phi\ (\{n_2, HPWD, ERN\} || Y_B), C \equiv \#(SK)$

$C|(S_1 \sim \{\{K_{C,CAS}, n_2, HPWD, ERN\}_{SK}, Y_B\}, C|, S_1 \ni SK$

The below given conditions are holds: 1) If C receives the rationale $\{K_{C,CAS}, n_2, HPWD, ERN\}$ 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 S_1 sent message is fresh. Therefore, the client module C validates the authentication received from the server S_1, if matched, and then client module believes that the client is legitimated entity.

If the client module C believes that the TTPA server S_1 sent n_2, HPWD, ERN, and 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 S_2 communications:

$C|\equiv \Phi\{\{K_{C,CAS}, n_2, HPWD, ERN\}_{SK}, \{K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP\}, Y_B\}\ S_1, C \ni SK,\ C \equiv S_1|\equiv \#(SK),\ S_2 \ni K_{CAS}$

$C|\equiv \Phi\{\{K_{C,CAS}, n_2, HPWD, ERN\}_{SK}, \{K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP\}, Y_B\}$

According to the proposed belief logic, the client module C believes that the TTPA authentication server is honest. We assumes $C|rS_1 => S_1|>_*$ and we form the following logical postulates for further adoption:

$C|aS_1 => S_1|>_*, C|Cn\{\{K_{C,CAS}, n_2, HPWD, ERN\}_{SK}, Y_B\},\ C|, S_1 \sim \{\{K_{C,CAS}, n_2, HPWD, ERN\}_{SK}, Y_B\}, S_1|, C \leftrightarrow S_1$

$$C \equiv S_1|\equiv C \leftrightarrow S_1$$

6.3 The Third Flow

$S_2 \lhd \{\{K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP\}, \{ID_{CS}, NA_C, n_3, HFP\}\},$

$S_1, S_2 \ni K_{CAS}, S_2, C \ni K_{C,CAS}$

$S_2 \lhd \{K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP\}$

If the TTPA server S_2 is informed by the client C that the message $\{K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP\}$ is encrypted with a TTPA AS and CAS shared Secrete Key K_{CAS}, then S_2 obtains

$\{K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP\}$ using corresponding Secrete Key K_{CAS}. From the received and decrypted message.

$$S_2|\equiv\Phi\ (K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP), S_1, S_2 \ni K_{CAS}$$

$$S_2|\equiv\Phi\ \{K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP\}$$

The below given conditions are holds: 1) If S_2 receives the rationale $\{K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP\}$ encrypted with K_{CAS}; 2) S_2 believes that all the rationale components received are recognizable; 3) S_2 decrypts rationale $\{K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP\}$ using its shared Secrete Key K_{CAS}; 4) S_2 trusts that $K_{C,CAS}$ is fresh one-time session key and used for decrypting client authentication details; 5) S_2 entitled to trust that C and S_1 sent message are fresh. Therefore, the cloud authentication server S_2 validates the authentication details received from the server S_1 and C, if matched, then S_2 believes that the client has authenticated entity.

If S_2 believes that the client $K_{C,CAS}, ID_{CS}, NA_C, n_2, HFP$ are recognizable, then $K_{C,CAS}$ will be used to decrypt the user details and then verifies these details with S_1 sent details. If ID_{CS}, NA_C, nonce and HFP are matched, then S_2 accepts the client as authenticated and allow to access the cloud services. Therefore, we can understand that the cloud authentication server S_2 believes TTPA server S_1 sent details for validating the user authentication details.

$$S_2|\equiv\Phi\ \{\{ID_{CS}, NA_C, n_3, HFP\}, C|\equiv S_2 \equiv C \leftrightarrow S_2$$

$$S_2|\equiv\Phi\ \{ID_{CS}, NA_C, n_3, HFP\}$$

6.4 The Fourth Flow

$$C \lhd \{\{K_{C,CARS}, n_4, OTP\}, \{K_{C,CARS}, ID_{CS}, NA_C, n_4,$$

$$OTP\}\}, S_3, C \ni K_{C,CARS}, S_2, C \ni K_{C,CAS}, S_2, S_3 \ni K_{CARS}$$

$$C \lhd \{\{K_{C,CARS}, n_4, OTP\}, \{K_{C,CARS}, ID_{CS}, NA_C, n_4, OTP\}\}$$

If the client C is informed by the cloud authentication server S_2 that the message $\{K_{C,CARS}, n_4, OTP\}$is encrypted with the one-time session key $K_{C,CAS}$, then C can obtain $\{K_{C,CARS}, n_4, OTP\}$ using $K_{C,CAS}$. From the received message, the client's decrypted contents.

$$C|\equiv\Phi\ (K_{C,CARS}, n_4, OTP), S_2, C \ni K_{C,CAS}$$

$$C|\equiv\Phi\ \{K_{C,CARS}, n_4, OTP\}$$

If the client module C feels that $K_{C,CARS}$, n_4 and OTP are recognizable and then C checks the nonce, if it is valid, then entitled to believe that the rationale parameters $K_{C,CARS}$, n_4 and OTP are fresh. Therefore, C believes that the received credential parameters are fresh.

$$C|\equiv\Phi\ \{\{K_{C,CARS}, n_4, OTP\}, \{K_{C,CARS}, ID_{CS}, NA_C, n_4,$$

$$OTP\}\}, S_3, C \ni K_{C,CARS}, C|\equiv S_2 \equiv C \leftrightarrow S_2, S_2, S_3 \ni K_{CARS}$$

$$C|\equiv\Phi\ \{\{K_{C,CARS}, n_4, OTP\}, \{K_{C,CARS}, ID_{CS}, NA_C, n_4, OTP\}\}$$

The below given conditions are holds: 1) If C receives the rationale component $\{K_{C,CARS}, n_4, OTP\}$ encrypted with $K_{C,CAS}$; 2) C believes that all the rationale components received are recognizable; 3) C obtains $\{K_{C,CARS}, n_4, OTP\}$ using $K_{C,CAS}$; 4) C trusts that $K_{C,CAS}$ is fresh temporal session key for the fourth flow; 5) C entitled to trust that S_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_2 sent rationale components are recognizable and content are matched, then accepts the client and considers OTP and $K_{C,CARS}$ for further user authorization process at cloud authorization server. Therefore, we can understand that the client module trusts and continues the client and S_3 communications.

$$C|\equiv\Phi\ \{\{K_{C,CARS}, n_4, OTP\}, \{K_{C,CARS}, ID_{CS}, NA_C, n_4, OTP\}\}$$
$$S_2, C \ni K_{C,CAS}, S_3 \ni K_{CARS}$$

$$C|AR\{\{K_{C,CARS}, n_4, OTP\}\}$$

According to the proposed belief logic, the client module C believes that the cloud authentication server is honest. We assumes $C|cS_2 \Rightarrow S_2|>_*$ and we form the following logical postulates for further adoption.

$$C|aS_2 \Rightarrow S_2|>_*, C|C\ \#\{\{K_{C,CARS}, n_4, OTP\}\}, C|$$
$$S_2 \sim \{\{K_{C,CARS}, n_4, OTP\}\}, S_2|, C \leftrightarrow S_2$$

$$C \equiv S_2|\equiv C \leftrightarrow S_2$$

6.5 The Fifth Flow

$$S_3 \lhd \{\{K_{C,CARS}, ID_{CS}, NA_C, n_4, OTP\}, \{ID_{CS}, NA_C, n_5, OTP\}\}, S_2,$$
$$S_3 \ni K_{CARS}, S_3, C \ni K_{C,CARS}$$

$$S_2 \lhd \{K_{C,CARS}, ID_{CS}, NA_C, n_4, OTP\}$$

If the cloud authorization server S_3 is informed by the client C that the message $\{K_{C,CARS}, ID_{CS}, NA_C, n_4, OTP\}$ is encrypted with a CARS and CAS shared secrete key K_{CARS}, then S_3 obtains $\{K_{C,CARS}, ID_{CS}, NA_C, n_4, OTP\}$ using

corresponding Secrete Key K_{CARS}. From the received and decrypted message.

$$S_2|\equiv\Phi\ (K_{C,CARS},ID_{CS},NA_C,n_4,OTP),S_2,\ S_3\ni K_{CARS}$$

$$S_2|\equiv\Phi\ \{K_{C,CARS},ID_{CS},NA_C,n_4,\ OTP\}$$

The below given conditions are holds: 1) If S_3 receives the rationale $\{K_{C,\ CARS},\ ID_{CS},\ NA_C,\ n_4,\ OTP\}$ encrypted with K_{CARS}; 2) S_3 believes that all the rationale components received are recognizable; 3) S_3 decrypts rationale $\{K_{C,\ CARS},\ ID_{CS},\ NA_C,\ n_4,\ OTP\}$ using its shared Secrete Key K_{CARS}; 4) S_3 trusts that $K_{C,CARS}$ 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_2 and C, if matched, then believes that the client has authorized entity.

If S_3 believes that the client ID_{CS}, NA_C, n_5 and OTP are recognizable, then $K_{C,CARS}$ will be used to decrypt the user details and then verifies these details with S_2 sent details. If ID_{CS}, NA_C, n_5 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.

$$S_3|\equiv\Phi\ \{\{ID_{CS},\ NA_C,\ n_5,\ OTP\},\ C|\equiv S_3\equiv C\leftrightarrow S_3$$

$$S_3|\equiv\Phi\ \{ID_{CS},\ NA_C,\ n_5,\ OTP\}$$

6.6 The Sixth Flow

$$C\lhd\{n_6\}$$

$$C\lhd\{n_6\}$$

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 $K_{C,CARS}$, then C can obtain $\{n_6\}$ using $K_{C,CARS}$. From the received message, the client's decrypted contents.

$$C|\equiv\Phi\ (n_{6),}S_3,C\ni K_{C,CARS}$$

$$C|\equiv\Phi\ \{n_{6\}}$$

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.

$$C|\equiv\Phi\ \{n6\}S_3,C\ni K_{C,CARS},\ C|\equiv S_3\equiv C\leftrightarrow S_3$$

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

The below given conditions are holds: 1) If C receives the rationale component $\{n_6\}$encrypted with $K_{C,\ CARS}$; 2) C believes that the received rationale is recognizable; 3) C decrypts $\{n_6\}$ using $K_{C,\ CARS}$; 4) C trusts that $K_{C,\ CARS}$ 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.

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

$$C|,C\{n_6\}$$

According to the proposed belief logic, the client module C believes that the cloud authorization server is honest. We assumes $C|cS_3{=}{>}S_3|{>}_*$ and we form the following logical postulates for further adoption.

$$C|aS_3{=}{>}S_3|{>}_*,C|C\ \#\{\{n_6\}\},\ C|\ S_3{\sim}\{\{n_6\}\ \},S_3|,C\leftrightarrow S_3$$

$$C\equiv S_3|\ C\leftrightarrow S_3$$

7. Expeimental 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[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 x 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[13].

1.1.3 Performance Evaluation

In our approach we used elliptic curve cryptosystem[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[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[18,22] mechanisms authentication servers decide UID's and passwords for remote users. The solutions[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 times and also avoids masquerade, eavesdrop and other replay attacks. In[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[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 3. Details of sensors used in Fvc2006

Data base	Sensor Type	Resolution	Image Size
FDB1	Optical	569 dpi	400x560(224Kpixels)
FDB2	Electric Field	250 dpi	96x96(9Kpixels)
FDB3	Thermal sweeping	500 dpi	400x560(200Kpixels)
FDB4	SFinGe v3.0	500 dpi	288x384(108Kpixels)

Table 4. Performance comparison

	C1	C2	C3	C4	C5	C6
A.Jyoti Choudhury et al.[18]	YES	NO	NO	NO	NO	NO
Ping Wang et al.[19]	NO	YES	YES	YES	YES	NO
B.Rohitash Kumar et al.[20]	YES	YES	NO	NO	NO	YES
Wenyi Liu et al.[21]	YES	YES	YES	NO	NO	YES
Hong Liu et al.[22]	NO	NO	YES	YES	YES	NO
Our Approach	YES	YES	YES	YES	YES	YES

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.

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 $t_p/(t_p+f_n)*100\%$, 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 $t_n/(t_n+f_p)*100\%$, 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,

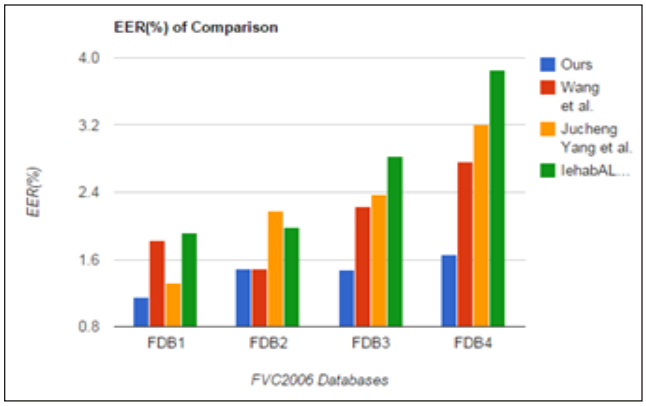

Figure 4. Our proposed approach recognition performance.

UID	HPWD	ERN	HBF	MN	p	g	Status
UID_1	$HPWD_1$	ERN_1	HBF_1	MN_1	p_1	g_1	Valid/ Invalid
UID_2	$HPWD_2$	ERN_2	HBF_2	MN_2	p_2	g_2	Valid/ Invalid
....	....	....	....	....	....	....	....
UID_i	$HPWD_i$	$ERNi$	$HBFi$	MN_i	p_i	g_i	Valid/ Invalid
....	....	....	....	....	....	....	....

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

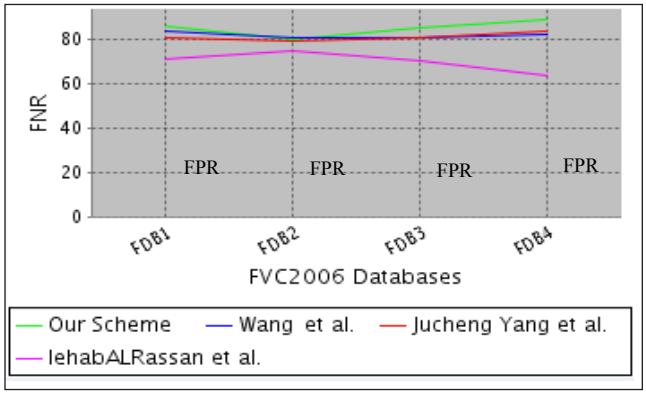

Figure 5. Comparison of false negative rates.

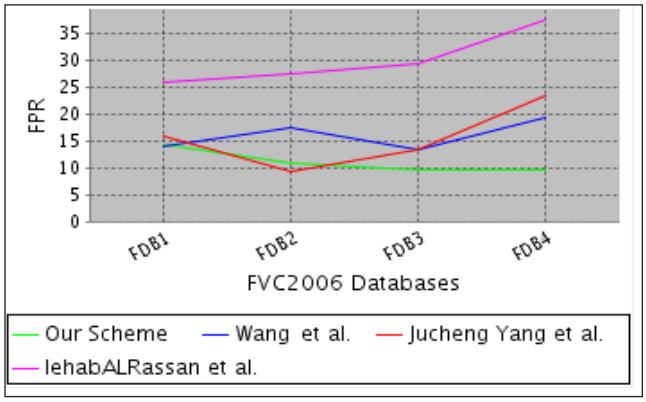

Figure 6. Comparison of false positive rates.

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

Data base	EER	REER	RE	RM	AET	AMT
FDB1	1.15%	1.15%	0.00%	0.00%	1.23 s	0.18 s
FDB2	1.49%	1.49%	0.00%	0.00%	1.53s	0.19 s
FDB3	1.48%	1.48%	0.00%	0.00%	1.74 s	0.14 s
FDB4	1.66%	1.66%	0.00%	0.00%	1.76 s	0.21 s
Avg.	1.44%	1.44%	0.00%	0.00%	1.56 s	

Figure 7. Equal error rate comparison study.

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[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[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[6,7]. In[6] pointed out that Lee's authentication approach cannot protect conspiring attack. Lin et al.[7] discovered that an authorized user can make any number of fake valid credentials to masquerade other authorised users. Lin et al.[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[15]. Yoon et al.[15] presented a solution to resist this attack. This improved solution was broken by Lee et al.[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. in[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[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.[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[19,35,36]. In[19], Ping Wang et al. described a secret-splitting authentication method for enhancing cloud security using smart-card. In this approach 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.[20] proposed a MFA framework using the OTP and IMEI number as authentication secretes. In[21], W. Liu et al. described a multi-factor cloud authentication approach using user password and secure user profile. However, the schemes[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.[22] discovered a privacy-preserving authentication scheme based on the

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

1. Highland HJ. Random bits and bytes: Testing a Password System. Computers and Security. 1989; 8:647–57.
2. Klein DV. Foiling the Cracker: A survey of and Improvements to Password Security. Proceedings of the 2nd USENIX UNIX Security Workshop; 1990. p. 1–11.
3. Morris R, Thompson K. Password Security: A case history. Communications of the ACM. 1979 Nov; 22(11):594–7.
4. Spafford EH. Observing reusable password choices. Proceedings of the 3rd UNIX Security Symposium; 1992. p. 299–312.
5. Lee JK, Ryu SR, Yoo KY. Fingerprint-based remote user authentication scheme using smart cards. Electronics-Letters. 2002 Jun; 38(12):554–5.
6. Chang CC, Lin IC. Remarks on fingerprint-based remote user authentication scheme using smart cards. ACM SIGOPS Operating Systems Review. 2004 Oct; 38(4):91–6.
7. Lin CH, Lai YY. A flexible bio-metrics remote user authentication scheme. Computers Standards Interfaces. 2004 Nov; 27(1):19–23.
8. Martino AS, Perramon X. A model for securing e-banking authentication process: Antiphishing approach. Proceedings of the 2008 IEEE Congress on Services; Honolulu, HI. 2008 Jul 6-11. p. 251–4.
9. Guan B, Wu Y, Wang Y. A novel security scheme for online banking based on virtual machine. Proceedings of the 2012 IEEE Sixth International Conference on Software Security and Reliability Companion; Gaithersburg, MD, USA. 2012 Jun 20-22. p. 12–7.
10. Musleh MMM, Ba II, Nofal KMA, Ibrahim J. Improving information security in e-banking by using Bio-metric Fingerprint. Proceedings of the International Journal of Computer Science and Information Security. 2012 Mar; 10(3):7–12.
11. Miller V. Uses of elliptic curves in cryptography. Advances in Cryptology—Crypto'85. Proceedings. Berlin Heidelberg: Springer-Verlag. 1986; 218:417–26.
12. Cappelli R, Ferrara M, Franco A, Maltoni D. Fingerprint verification competition 2006. Bio-metric Technology Today. 2007 Jul-Aug; 15(7-8):7–9.
13. GEDIS studio online test data generator. Available from: http://www.data-generator.com/
14. Uludag U, Pankanti S, Prabhakar S, Jain AK. Bio-metric cryptosystems: Issues and challenges. Proceedings of the IEEE. 2004 Jun; 92(6):948–60.
15. Yoon EJ, Yoo KY. A new efficient fingerprint-based remoteuser authentication scheme for multimedia systems. 9th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES 2005). 2005; 3683:332–8.

16. Lee Y, Kwon T. An improved fingerprint-based remote user authenticationscheme using smart cards. Proceedings on Computational Science and its Applications ICCSA'06. 2006; 3981:915–22.
17. Bhargav-Spantzel A, Squicciarini AC, Bertino E, Modi S, Young M, Elliott SJ. Privacy preserving multi-factor authentication with bio-metrics. Journal of Computer Security. 2007; 15(5):529–60.
18. Amlan AJ, Kumar P, Sain M, Lim H, Jae-Lee H. A strong user authentication framework for cloud computing. 2011 IEEE Asia-Pacific Services Computing Conference; Jeju Island. 2011 Dec 12-15. p. 110–5.
19. Wang P, Ku CC, Wang TC. A new fingerprint authentication scheme based on secret-splitting for enhanced cloud security. Recent Application in Bio-metrics; 2011. p. 183–96.
20. Banyal RK, Jain P, Jain VK. Multi-factor authentication framework for cloud computing. 2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation (CIMsim); Seoul. 2013 Sep 24-25. p. 105–10.
21. Liu W, Selcuk Uluagac A, Beyah R. MACA: A privacy-preserving multi-factor cloud authentication system utilizing big data. 2014 IEEE Conference on INFOCOM Workshop on Computer Communications, Infocin workshops; Toronto, ON. 2014. p. 518–23.
22. Liu H, Ning H, Xiong Q, Yang LT. Shared authority based privacy-preserving authentication protocol in cloud computing. IEEE Transactions on Parallel and Distributed Systems. 2015 Jan; 26(1):241–51.
23. Ramesh N. The Ultimate Guide for Creating Strong Passwords. 2008. Available from: http://www.thegeekstuff.com/2008/06/the-ultimate-guide-for-creating-strong-passwords/
24. Create strong passwords. 2014. Available from: https://www.microsoft.com/security/pc-security/password-checker.aspx
25. Six rules for safer financial transactions online. 2014. Available from: http://www.microsoft.com/security/online-privacy/finances-rules.aspx
26. Overcoming Security, Privacy and Compliance Concerns. White Paper. 2013 p. 1–13. Available from: www.ciphercloud.com
27. Burrows M, Abadi M, Needham R. A logic of authentication. ACM Transactions on Computer-Systems. 1990 Feb; 8(1):18–36.
28. Nessett DM. A critique of the Burrows, Abadi and Needham logic.Operating Systems-Review. 1990 Apr; 24(2):35–8.
29. Gong L, Needham R, Yahalom R. Reasoning about belief incryptographic protocols. Proc 1990 IEEE Computer Society Symposium on Research in Security and Privacy; Oakland, CA. 1990 May 7-9. p. 234–48.
30. Dodis Y, Ostrovsky R, Reyzin L, Smith A. Fuzzy extractors: How to generate strong keys from bio-metrics and other noisy data. Advances in Cryptology, Eurocrypt'04. 2004; 3027:523–40.
31. Uludag U, Pankanti S, Prabhakar S, Jain AK. Bio-metric cryptosystems: Issues and challenges. Proceedings of the IEEE, Special Issue on Multimedia Security for Digital Rights Management. 2004 Jun; 92(6):948–60.
32. Raphael JR. Last Pass CEO Explains Possible Hack. 2015. Available from: http://www.pcworld.com/article/227268/lastpass_ceo_exclusive_interview.html
33. Nagaraju S, Parthiban L. Achieving privacy protection of multi-factor authentication and access keys in cloud computing. Proceedings of 3rd National Conf on Frontiers in Applied Sciences And Computer Technology (FACT'15); 2015. p. 308–15.
34. Nagaraju S, Parthiban L, Santhosh Kumar B. Achieving privacy protection of multi-factor authentication and access keys in cloud computing. Proceedings of 3rd National Conference on Frontiers in Applied Sciences And Computer Technology (FACT'15); 2015. p. 308–16.