Safe Cloud: Secure and Usable Authentication Framework for Cloud Environment

Binu Sumitra, Pethuru Raj and M. Misbahuddin

Abstract Cloud computing an emerging computing model having its roots in grid and utility computing is gaining increasing attention of both the industry and laymen. The ready availability of storage, compute, and infrastructure services provides a potentially attractive option for business enterprises to process and store data without investing on computing infrastructure. The attractions of Cloud are accompanied by many concerns among which Data Security is the one that requires immediate attention. Strong user authentication mechanisms which prevent illegal access to Cloud services and resources are one of the core requirements to ensure secure access. This paper proposes a user authentication framework for Cloud which facilitates authentication by individual service providers as well as by a third party identity provider. The proposed two-factor authentication protocols uses password as the first factor and a Smart card or Mobile Phone as the second factor. The protocols are resistant to various known security attacks.

Keywords Cloud computing · Authentication issues · Single sign-on · SAML · Mobile authentication

B. Sumitra (✉)
Christ University, Bangalore, India
e-mail: sumitrabinu@gmail.com

P. Raj
IBM Cloud Global Center of Excellence, IBM India Pvt. Ltd., Bangalore, India
e-mail: peterindia@gmail.com

M. Misbahuddin
C-DAC, Electronic-City, Bangalore, India
e-mail: mdmisbahuddin@gmail.com

R. Chaki et al. (eds.), *Advanced Computing and Systems for Security*,
Advances in Intelligent Systems and Computing 395,
DOI 10.1007/978-81-322-2650-5_12

1 Introduction

The advent of Web 2.0 has contributed to an exponential growth in the users of Internet and related technologies. Cloud computing, an Internet-based distributed computing model offering computing resources as a service, is a fast growing technology slowly being embraced both by corporate sector as well as by laymen. With the advancements in IT, this technology has evolved through a number of different services such as software applications (SaaS), computing platform, (PaaS), and infrastructure (IaaS). Thus Cloud computing refers to both the applications delivered as services over the Internet and the hardware and system software in the data centers that provide those services [1]. The primary objective of Cloud computing is to provide great flexibility to users, by allowing the users to process, store and access their data using Cloud services, anytime, anywhere using the Internet without investing on Computing Infrastructure.

The fast growing utility-based Cloud computing technology offers many advantages such as resource sharing dynamic scalability, rapid elasticity, efficient software/platform/infrastructure utilization, and many more [2]. However, this technology has a lot of concerns including performance, resiliency, interoperability, data migration and transition from legacy systems, preventing the whole hearted adoption of Cloud services. Among the many issues, one of the most relevant is security, which involves virtualization security, distributed computing, application security, identity management, access control and authentication [3–5]. Furthermore in [6, 7] authors have pointed out that the identity and access control management is a core requirement for Cloud computing. Hence, strong user authentication which thwarts illegal access of Cloud servers becomes the fundamental requirement in the Cloud computing environment.

Over the last few years, significant research has happened in the security-related areas of Cloud computing with the objective of arriving at mechanisms that provide adequate security to Cloud environment and its users [2, 8–10]. However, these existing security mechanisms are susceptible to certain security attacks, when examined from the perspective of practical implementation. Many existing public Clouds such as Amazon Web Services, Dropbox, Salesforce.com, and Google App Engine, etc., have been victimized to various security attacks [11–14]. Hence it is possible for illegal users to exploit these security flaws and either steal secret information or disturb the normal operation of Cloud service providers by launching various attacks. This points out to the requirement for securing Cloud with a strong user authentication mechanism that prevents illegal users from performing various nefarious activities in the Cloud.

Authenticating the identity of remote users is a preliminary requirement in a public Cloud environment before they can access a secure resource. Service providers (SP) should ensure that only authorized users are accessing services provided by the application system and password authentication is one of the simplest and most convenient user authentication mechanism. Unfortunately, users tend to choose low entropy passwords which are easy to remember rendering the authentication

system susceptible to various attacks. Strong authentication mechanisms address this issue by authenticating users based on a combination of two or more factors, viz., what you know, what you have, and what you are.

Taking into consideration, the security issues faced by Cloud, the discussed work proposes a strong and reliable two-factor user authentication framework for Cloud environment. The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 discusses novelty of our contribution, Sect. 4 explains the authentication framework, Sects. 5 and 6 elaborate the authentication architectures and protocols respectively and Sect. 7 concludes the work done.

2 Related Work

Choudhary et al. proposed a user authentication framework for Cloud environment [2] using password, Smart Card, and out-of-band (OOB) authentication token. Cloud environment comprises of multiple service providing servers and users may access services from servers belonging to different domains. The single sign-on (SSO) functionality which provides convenient user authentication in such a scenario was not considered by Choudhary et al. In 2013 Jiang [10] proved that the scheme [2] was prone to masquerade user attack, the OOB attack, and the password change flaw and proposed a modified scheme. The scheme addresses the security issues of [2], but uses time stamps which can lead to time synchronization problems. The protocol [15] stores a variant of the user password in the server which makes it susceptible to stolen verifier attack.

Sanjeet et al. [16] proposed a user authentication scheme using symmetric keys for Cloud services. The protocol uses a one-time token which is sent to the registered users e-mail ID. This scheme requires the user to login to two accounts during the authentication process which may cause user inconvenience. As in the case of [2], the authentication schemes proposed by Rui Jiang and Sanjeet et al. does not provide the SSO functionality which enhances user convenience in a multi-server Cloud environment.

3 Novelty of Our Contribution

Primary objective of the work is to design an authentication framework for cloud services encompassing authentication models and authentication protocols, which can be used by two categories of organizations, namely collaborative organizations and financial institutions. The two-factor authentication protocols are designed to overcome the limitations of currently prevalent mechanisms and be capable of operating in a traditional environment as well as in a Smart environment. The authentication model addresses the issues related to storing passwords at the cloud service provider's end.

The proposed lightweight authentication protocols are designed to have minimum processing overhead and to be resistant to the common attacks on authentication.

4 User Authentication Framework for Cloud

This section provides an overview of the proposed authentication framework for Cloud and discusses some approaches to address its components. The overall authentication framework and the key components required to integrate and provide services in a secure manner are depicted in Fig. 1. The proposed framework comprises of three major entities, viz., the users, the cloud brokers (CBs), and the cloud service providers (CSPs). The CBs act as an identity provider (IdP) and as an intermediary between the users and CSPs. They work with the users to understand the work processes, provisioning needs, budgeting, data management requirements, etc. The CBs then discover services from different CSPs or other brokers, carry out negotiations, integrate the services to form a group of collaborating services and recommend the concerned CSPs to the user. Each CB has components that are

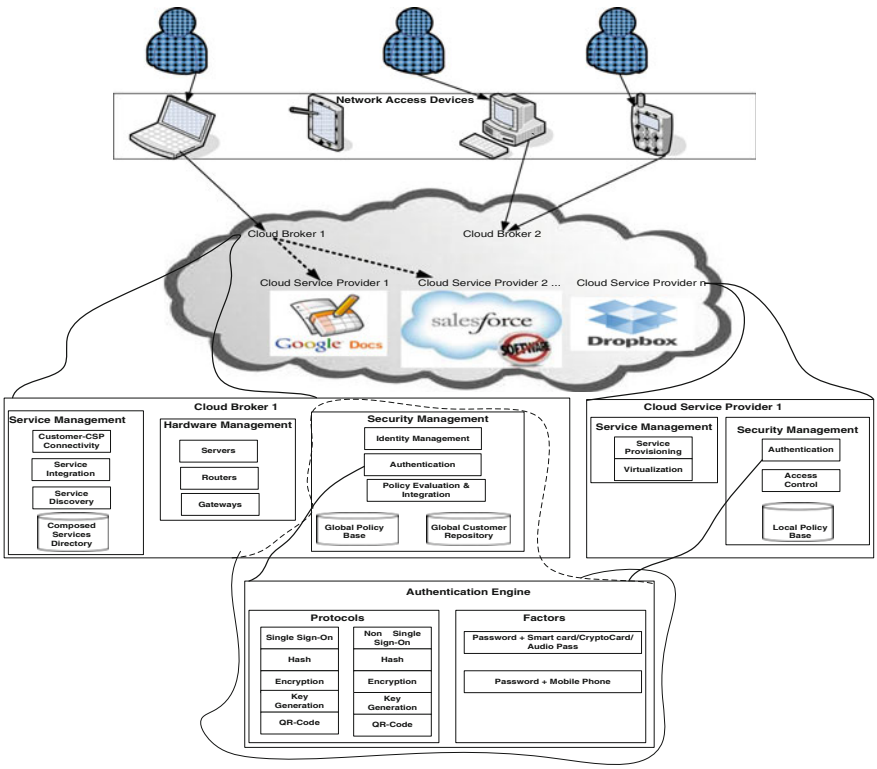

Fig. 1 Authentication framework for cloud environment

responsible for ensuring security and establishing trust between local provider domains and between the providers and users as well as generating global policies. In a Cloud environment, users should have some level of trust on each other.

4.1 Components of the Framework

This section details the functionality of the various components of Cloud Brokers and the CSPs of the framework.

Service Management This component of CB is responsible for secure service discovery by interacting with the CSPs, integrating the services, composing new desirable services, and establishing the link between the user and the CSPs [17]. Discovery of services, integrating the services, and connecting the users and CSPs are done by the service discovery module, service composition module and the customer–CSP connectivity module respectively. The Service Management component of the CSP is responsible for provisioning the services to the user. To support multi-tenancy and resource sharing, the CSP uses Virtualization technology.

Security Management The Security Management component comprises of the modules whose functionality aids in enforcing security and trust. The identity management module of CB is responsible for issuing and managing the identification credentials of the users registered with the CB. The authentication module is responsible for authenticating users and CSPs based on the credentials. The authentication module of CB executes a two-factor authentication protocol and uses SAML to provide user convenience through Single Sign-On functionality. The CSPs may have conflicting interests regarding the policies they adopt to provide services to their users and this may be a matter of concern when multiple CSPs collaborate to provide a customized service. Specification frameworks are needed to ensure that the cross domain accesses are properly specified and enforced [17]. SAML, XACML, and WS standards are viable solutions toward these needs and the proposed framework uses to address this requirement. The policy evaluation and integration module examines the policies of various CSPs whose services need to be integrated. The module then addresses security challenges such as semantic heterogeneity, secure interoperability, and integrate access policies of different CSPs and define global policies to accommodate the requirements of all the collaborating CSPs. These global policies are available in the global policy repository. The security management component also includes a global customer repository that stores the details of the registered CSPs and users.

Authentication Engine The functionality of the authentication engine is accessed by the authentication module of both the CBs and the CSPs to authenticate the users prior to providing access to the services provided by the CSPs. The proposed authentication protocols verify user authenticity by a two-factor authentication mechanism, and do not require the authentication server to maintain a verification

table. The scheme uses password as the first factor and a Smart Card/Crypto Card/Audio Pass/Mobile Phone as the second factor. The advantage of using an Audio Pass is that it can be used with smart phones and hence do not require an additional device (Card reader) to read the stored data as in the case of Smart Cards/Crypto Cards. The paper proposes two different authentication architectures, one in which the IdP/CB authenticates the users prior to accessing the services of CSP and thus provides a SSO facility. In the second authentication the users are authenticated by the CSPs whose service they wish to access. In both the architectures both the users and the CSPs need to first register with the IdP/CB. The authentication protocols supported by both the architectures provide two-factor authentication by using password as the first factor and Smart Card/Crypto Card/Audio Pass/Mobile Phone as the second authentication factor. Smart Card/Crypto Card/Audio Pass/USB Token is primarily meant to be used in the case of those departments/organizations/users for which security is the top priority and hence the additional hardware cost and the extra inconvenience is acceptable. Mobile phone as a second authentication factor is targeting on those departments/organizations who would like to ensure security by making use of a commodity that the user already owns rather than burdening the user with an additional hardware [18].

5 Authentication Architectures

The following sections give a brief overview of the authentication architectures proposed in this work.

5.1 Broker-Based Authentication Architecture

The Coca-Cola company is collaborating with Heinz to use their bottling factory to make PET bottles using 100 % plant-derived materials and plant residues. To effectively reuse the used bottles, the Coca-Cola Company and furniture company Emeco have entered into a collaboration to make Emeco 111 navy chair, a chair made of 111 recycled bottles. Similarly, Biotherm, a skin care company and the automobile manufacture Renault has collaborated to invent the new concept in cars, 'The Spa Car.' These collaborative organizations can offer their services via the public cloud which offers the advantages of resource sharing, standardization of operations, increased reusability, reduced capital expenditure, etc. These collaborative organizations will be having a common customer base who would like to access the services and information from all these organizations. When these individual organizations offer their services from a cloud environment, each of them will have their own applications server to provide the services and the information. In a conventional environment, a customer who wants to access services of all these collaborating organizations will be required to create individual accounts with each

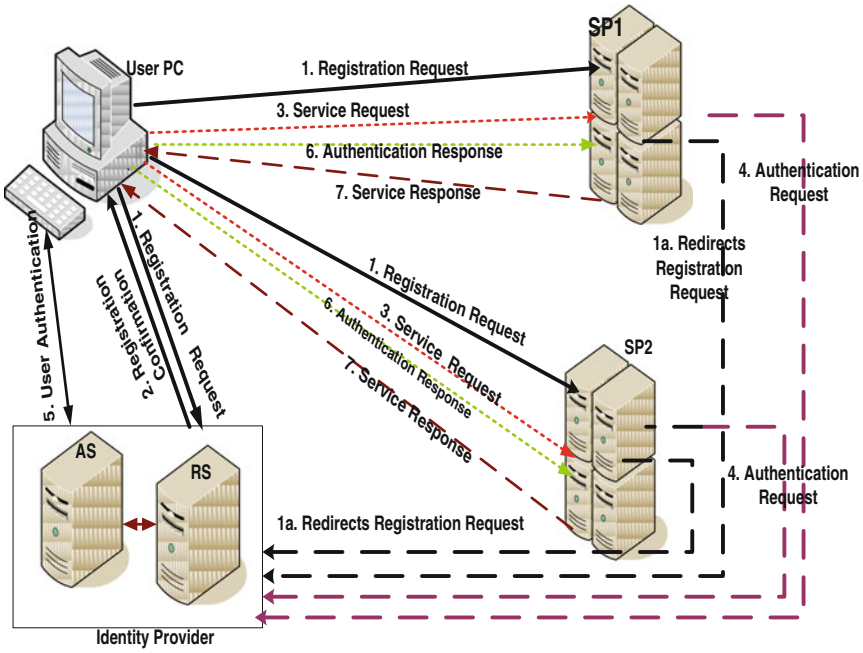

Fig. 2 Registration and authentication process flow in broker-based authentication architecture

service provider. This contributes to multiple accounts, multiple passwords and multiple authentication to access multiple services. Nevertheless these organizations would prefer to provide their customers with a user-friendly procedure that enables them to access information from all the collaborating partners with ease. The work proposes the use of a single account to access the services of multiple service providers and a user-friendly login process that allows the user to authenticate once and access multiple services during a single login session, termed as single sign-on. To support single account and single sign-on, we propose a broker-based architecture comprising of a centralized registration authority alias identity provider (IdP) and the multiple service providers. All the service providers and the users accessing the services of these service providers should be registered with the registration server of the centralized registration authority alias IdP. The users can register once and create an account at the IdP to get the services of all the registered collaborating service providers. Once registration is done user is issued an authentication token such as a Smart Card/Crypto Card/Audio Pass or he is given an option to download a secret file into his Mobile Phone if he is using Mobile Phone as the second authentication factor. To facilitate single sign-on, the authentication of users is also done by the authentication module of IdP who is trusted by the collaborating service providers. During login process, the service providers will redirect the users to the IdP who will authenticate the users using the proposed two-factor authentication protocol and send the response (token/assertion) to the service provider.

Thus user can access the services seamlessly and the services providers having handed over the responsibility of authenticating their customers to the IdP can concentrate on providing their core services. The registration and authentication process is depicted in Fig. 2. The registration phase includes registration server (RS), CSP, and users'. The RS and AS are in the same trusted domain and together they provide the functionality of the identity provider (IdP/CB). The CSPs and IdP work in a trust-based environment. The proposed architecture inherits all the desired features of a MSE and uses Security Assertion Markup Language (SAML) to provide user convenience through SSO [19].

5.2 *Direct Authentication-Based Architecture*

Financial institutions including commercial banks, investment banks, brokerages, insurance companies, etc., deal with highly sensitive data and hence reliable customer authentication is imperative for engaging in any form of electronic banking. These organizations when they move into the cloud would like to be assured that their data and information stored in the cloud is completely secure from unauthorized access. A strong and effective authentication system can help financial institutions to reduce fraud, to enforce anti-money laundering practices, and detect and reduce identity theft. The risk of engaging in business with unauthorized individual in a money-issuing environment could result in financial loss and reputation damage through fraud, disclosure of confidential information, corruption of data, etc. Considering these risks, these financial institutions when they operate from the cloud will not be willing to trust anybody regarding the authenticity of the users attempting to access their data and services and would prefer to have an internal mechanism for authentication. To address such a scenario, the work proposes a direct authentication

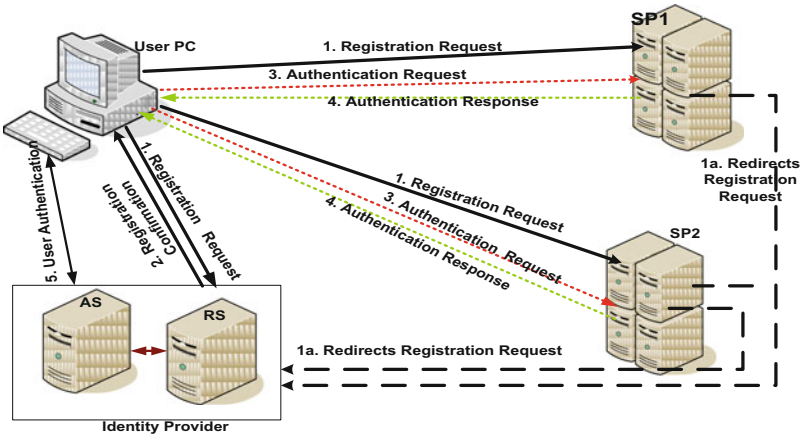

Fig. 3 Registration and authentication process flow in direct authentication-based architecture

architecture as shown in Fig. 3, where we place an authentication server in front of each service providing server belonging to the category of financial institutions. In this architecture, the users register themselves at the Identity Provider who issues them with an authentication token as in the case of broker-based architecture. After registering, a user who wants to access the services would attempt to login to the server of the financial institution and these servers will be independently authenticating their users using the proposed two-factor authentication protocol.

6 Authentication Protocols

The authentication protocols of the proposed work are categorized into two broad categories, viz., protocol for broker-based authentication architecture and protocol for Direct Authentication-based architecture. Again the two broad categories are subdivided into the sub categories (i) Protocol for broker-based authentication using password as the first factor and Smart Card/Crypto Card/Audio Pass as the second factor (ii) Protocol for broker-based authentication using mobile phone as the second factor (iii) Protocol for direct authentication using password as the first factor and Smart Card/Crypto Card/Audio Pass as the second factor (iv) Protocol for direct authentication using mobile phone as the second factor. This section discusses the various phases of the first three categories.

6.1 Protocol for Broker-Based Authentication Using Password as the First Factor and Smart Card as the Second Factor

Phases of the Protocol The proposed scheme consists of four phases, viz., Initialization phase, User registration phase, Login and Authentication phase, and Password change phase. The notations used in the protocol are listed in Table 1.

Table 1 Notations used in broker-based authentication using smart cards

AP, SC	Audio pass, smart card
U_a, IdP, SP	User 'a', identity provider, service provider
ID_a, P_a	Identity, password of user U_a
SID, y	Server ID of IdP, secret key of IdP
G	Additive cyclic group of prime order
g_0	Generator of additive cyclic group
r	Random number generated by audi pass unique to each session
$h(.)$, $\oplus$, $\|$	Hash function, XOR operation, concatenation operation
$\Rightarrow$	Secure communication channel

Initialization Phase During this phase, User U_a generates a finite additive cyclic group 'G' of prime order 'n' and selects an element 'g_0' from the group. 'g_0' is one of the generators of 'G' and is used by U_a to modify the password to be used for secure user registration and authentication.

User Registration Phase If user wants to register for the services of a Service Provider SP, the user U_a clicks the 'Create Account' link at SPs web site. SP redirects U_a to the registration page of the IdP. IdP prompts U_a to submit the Identity and Password of the user. U_a chooses her identity ID_a and Password P_a and the phase proceeds as illustrated in Fig. 4, which can be explained as follows.
UR1: U_a Computes $b = h(P_a)$, $k = g_0^b$ and submits $h(ID_a)$, k to IdP through a secure channel. IdP checks whether the submitted $h(ID_a)$ already exists in its user table and if so prompts U_a to submit a new ID, otherwise IdP proceeds as follows:

IdP computes $E_i = B_i \oplus h\,(\mathrm{SID}\|h(y))$ where 'y' is a secret key of IdP and $h(.)$ is a one way hash function. $B_i = h(h(ID_a)\|h(\mathrm{SID}))$; $J_i = h(\mathrm{SID}\|h(y)) \oplus k$; $C_i = h(h(ID_a) \| h(\mathrm{SID}\|h(y))\| k)$;

UR2: IdP personalizes the smart card (SC) with the parameters C_i, E_i, J_i, $h(.)$. IdP sends the SC to U_a via a secure channel.

UR3: On receiving the device, U_a stores g_0 into the Audio Pass/Smart Card/USB token which now contains $\{C_i, E_i, J_i, h(.), g_0\}$.

Login and Authentication Phase This section discusses the Cloud Service Provider initiated authentication and Fig. 5 gives a pictorial representation of the same.

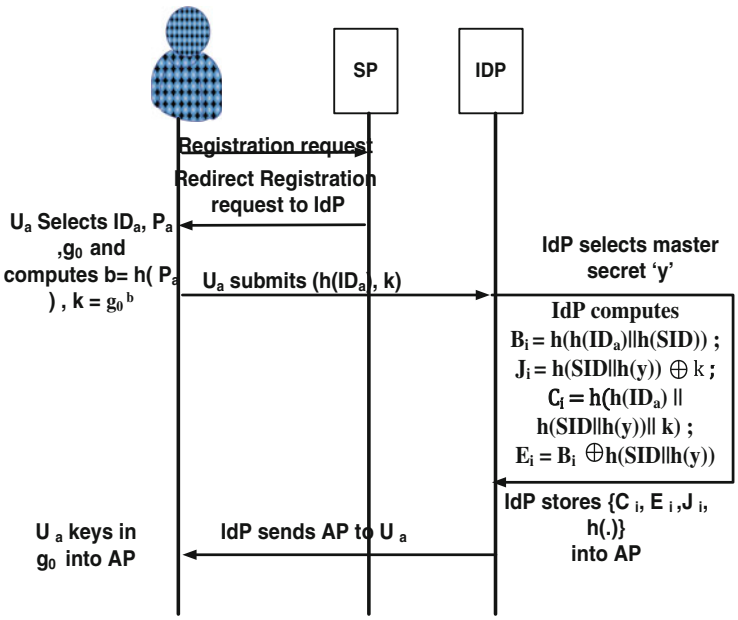

Fig. 4 Registration phase of broker-based authentication protocol using smart card/audio pass

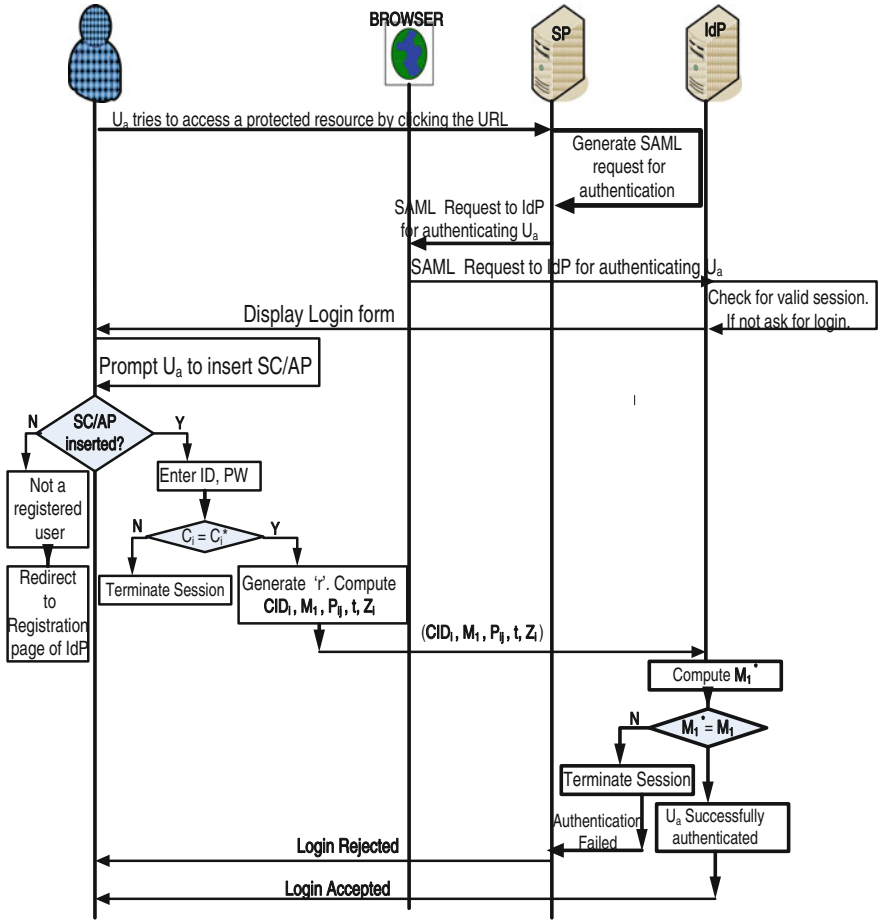

Fig. 5 Login and authentication phase of broker-based authentication protocol using smart card

Whenever a registered user wants to login to access the services of the Service Provider 'SP', she attaches the Smart Card or Audio Pass token to the system and proceeds as follows:

UL1: U_a requests for login to the SP

UL2: SP checks for an existing session with U_a and if there is no valid session, SP redirects U_a to IdP with a SAML authentication request

UL3: U_a keys in her ID_a and P_a

UL4: SC/AP computes, $b = h(P_a)$, $k^* = g_0^b$

UL5: SC/AP computes $h(\mathrm{SID}|h(y))^* = J_i \oplus k^*$

UL6: SC/AP computes $C_i{}^* = h(h(\mathrm{ID}_a) \parallel h(\mathrm{SID}\|h(y))^*\| k^*)$ and compares with C_i stored in the AP. If invalid, AP terminates the session. Otherwise generates the login message as follows:

UL7: AP generates a random number 'r' and computes nonce $n_1 = g_0^r$

UL8: AP computes $P_{ij} = E_i \oplus h(h(\text{SID}|h(y))\|n_1)$; $B_i = E_i \oplus h(\text{SID}\|h(y))$; $\text{CID}_i = C_i \oplus h(B_i\|n_1\|\text{SID})$; $M_1 = h(P_{ij}\ |C_i\ \|B_i\|n_1)$; $t = g_0 \oplus h(\text{SID}\|h(y))$; $Z_i = (r - \text{CID}_i) \oplus h(\text{SID}\|h(y))$

UL9: AP sends (CID_i, M_1, P_{ij}, t, Z_i) to IdP

UL10: Upon receipt of the login message the IdP performs the authentication process using her own SID and $h(y)$ values

UL11: IdP computes, $r = (Z_i + \text{CID}_i) \oplus h(\text{SID}\|h(y))$; $g_0 = t \oplus h(\text{SID}\|h(y))$; $n_1* = g_0^r$, $E_i = P_{ij} \oplus h(h(\text{SID}\|h(y))\|n_1)$; $B_i* = E_i \oplus h(\text{SID}\|h(y))$; $C_i* = \text{CID}_i \oplus h(B_i*\|n_1*\|\text{SID})$;

UL12: IdP computes $M_1 * = h(P_{ij}\ \|C_i*\ \|B_i*\|n_1*)$ and compares with the M_1 in the login message received from U_a. If valid, IdP considers the authentication as successful and creates a SAML authentication response message containing the result of the authentication process and redirects it to the SP. The Service Provider permits or denies access to the services after verifying the response from the IdP.

Password Change Phase A registered user can change her password by selecting the change password option and the password can be modified at the client side without the intervention of IdP and SP. The change password request is processed only if the keyed in ID and password are valid. This phase proceeds as follows:

UP1: U_a attaches his SC or AP into the system and keys in his ID_a and P_a

UP2: SC/AP computes, $b = h\ (P_a)$, $k = g_0^b$

UP3: SC/AP computes $h(\text{SID}\|h(y)) = J_i \oplus k$

UP4: SC/AP computes $C_i * = h(h\ (\text{ID}_a)\ \|\ h(\text{SID}\|h(y))*\|\ k*)$ and compares with C_i stored in the SC/AP. If invalid, SC/AP terminates the session. Otherwise prompts U_a to enter the new password

UP5: U_a enters $P_{a\text{new}}$

UP6: SC/AP computes $b_{\text{new}} = h(P_{a\text{new}})$; $k_{\text{new}} = g_0^{b\text{new;}}$ $J_{i\text{new}} = J_i \oplus k \oplus k_{\text{new}}$; $C_{i\text{new}} = h(h(\text{ID}_a)\ \|\ (J_i \oplus k)\|\ k_{\text{new}})$;

UP7: SC/AP replaces C_i and J_i in the AP/SC with $C_{i\text{new}}$ and $J_{i\text{new}}$ respectively.

Security Analysis of the Protocol This section analyzes the security of the proposed protocol against various attacks.

Security Against Guessing Attack The proposed protocol is secure against guessing attack as it is impossible within polynomial time, for an adversary to retrieve user's password P_a or IdP's secret key 'from the intercepted parameters (CID_i, M_1, P_{ij}, t, Z_i).'

Security Against Malicious Insider Attack In the proposed scheme, user submits $k = g_0^{h(\text{Pa})}$ to IdP rather than the plain text form of the password. This guards the password from being revealed to IdP and hence even if the user uses the same password to login to other servers, her credentials will not be susceptible to insider attack.

Security Against Replay Attack A replay attack is launched by the adversary by capturing a message exchanged between the Client and Server and replaying at a later point in time. The scheme is resistant to replay attack since nonce values used

to in each authentication message is unique and varies for each session. Hence the IdP will be able to identify a replayed login message (CID_i, M_1, P_{ij}, t, Z_i) by checking the freshness of nonce, $n_1 = g_0^r$ where 'r' is a random number generated by user and is unique to a session.

Security Against Stolen Verifier Attack The fact that the proposed scheme does not require the Server to maintain a verifier/password table makes it resistant to Stolen Verifier attack.

Security Against User Impersonation Attack If an adversary attempts to impersonate a valid user, he should be able to forge a valid login request on behalf of the user. In the proposed scheme if an adversary intercepts the login message (CID_i, M_1, P_{ij}, t, Z_i) and attempts to generate a similar message, he will fail since the value of nonce 'n_1' as well as the server's secret key 'y' is unknown to him.

Security Against Denial-of-Service (DoS) Attack A DoS attack can be launched by an adversary by creating invalid login request messages and bombarding the server with the same. This attack can also be launched by an adversary who has got control over the server and is able to modify the user information stored in the server's database which in turn prevents the valid user from accessing the resources.

The first scenario will not work in the case of the proposed scheme, since it is impossible for the adversary to create valid login request messages without knowing the password. The validity of the password is checked at the client side before creating a login request. The second scenario is also not applicable in the proposed scheme since the server does not maintain a verifier/password table.

Security Against Smart Card Lost Attack If the adversary steals the Smart Card containing the parameters (C_i, E_i, J_i, $h(.)$, g_0), he can neither retrieve the user's password nor the IdP's master secret 'y' from the stored value. To extract the password from $k = g_0^{h(\mathrm{Pa})}$, the adversary needs to solve the discrete logarithm problem. Again the password is used in the hashed form which is irreversible. Also, retrieving the IdP's master secret, 'y' is not possible without knowing the password of the user which is unknown to the adversary.

6.2 Protocol for Broker-Based Authentication Using Password as the First Factor and Mobile Phone as the Second Factor

This protocol consists of a registration phase, login and authentication phase, and a password change phase as elaborated in the following subsections. Here, it is assumed that the CSP who wants to be a part of the framework is registered with the IdP.

Phases of the Proposed Protocol The proposed scheme consists of three phases, viz., User registration phase, Login and Authentication phase, and Password change phase. The notations used are listed in Table 2.

Table 2 Notations used in broker-based authentication using mobile phone

ID_a, P_a	Identity, password of user U_a
s, rand	Secret key, random number of RS
r, Key_a, V_a, m	Random number generated by AS unique to each session, master secret key of user, parameter used for verifying the password, parameter used for changing the password
$h(.)$, $\oplus$, $\|$	Hash function, XOR operation, concatenation operation
$\Rightarrow$	Secure communication channel

Registration Phase The registration process illustrated in Fig. 6 can be explained as follows:

UR1. The user U_a clicks the 'Create Account' link at SPs web site. SP redirects U_a to the registration page of the IdP. RS prompts U_a to submit her identity ID_a and PW_a.

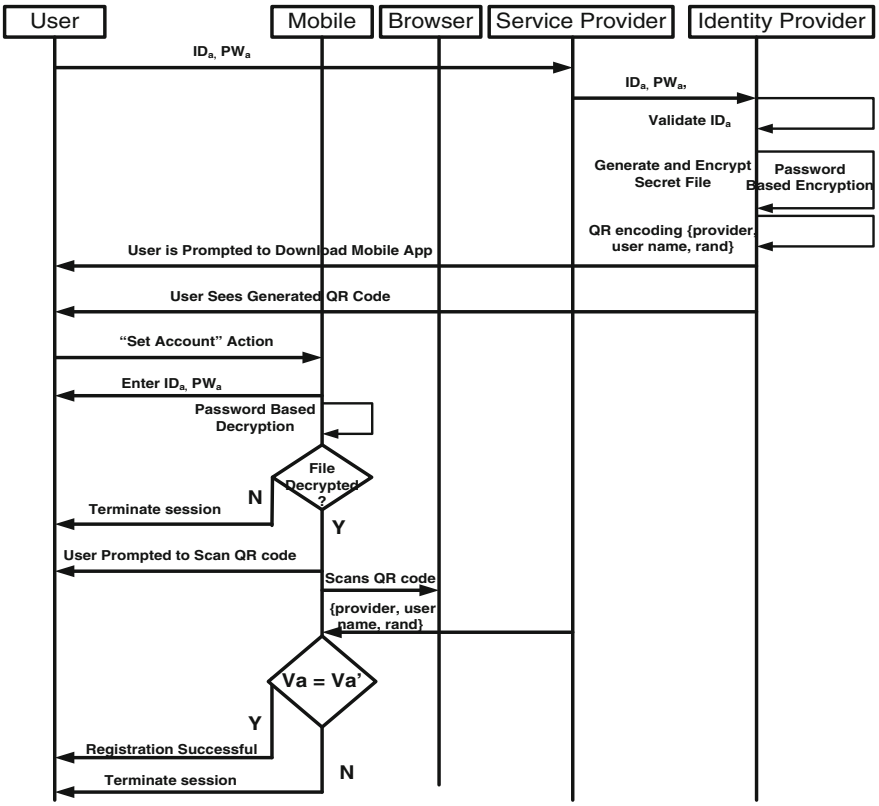

Fig. 6 Registration phase of broker-based authentication protocol using mobile phones

RS checks whether ID_a already exists in its user table. If so U_a is prompted to select a new ID_a.

UR2. RS generates a secret key 'S' and selects a random number 'rand'. RS creates a file containing the authentication parameters V_a, Key_a, m and the file is encrypted using password of U_a and a salt value. 'V_a' is used only during the registration and installation of mobile application. 'Key_a' is used during the authentication phase and 'm' is used during the password change phase. The values of V_a, Key_a, m are generated by performing hash and XOR operations on ID_a, PW_a, S, and rand.

UR3. The RS generates a QR code embedding 'rand', Service Provider Name and User Name and the QR code will be displayed on the web application screen and the user will be prompted to download the mobile application. Along with the app, the secret file will also be imported and stored in a safe location within the user's phone in the form of a mobile token. When the user touches the register button in the mobile app, the user will be prompted to enter his ID_a, PW_a. The app attempts to decrypt the file using password given as input by the user and the salt value attached to the file.

UR4. If the decryption is successful, the app invokes the scanning application, and the user can scan the code. The mobile app retrieves 'rand' from QR code, calculates V_a' using password and 'rand'. V_a' is compared with V_a stored in the mobile token and if equal, the registration process is considered successful and the account is created. RS stores the user identity in its user table.

Login and Authentication Phase As shown in Fig. 7, the user via his browser attempts to access a protected resource of a Service Provider (SP). It is assumed that, the browser at this point does not have an established session with the SP. On receiving the request from the user, SP redirects the user to the login page of IdP and requests IdP to authenticate the user. The authentication request contains the URL of the SP who initiated the request. Also the request should contain the URL to which the response should be sent. IdP checks for a valid session with the browser. If there is no existing session between the browser and the IdP, then IdP generates a login session and authenticates the user by executing the authentication phase, as illustrated in Fig. 7. The procedure can be explained as follows:

UA1. Authentication server (AS) displays the login page and prompts the user to enter user's identity (ID_a) and Password (PW_a). AS calculates Key_a and a challenge 'B' using server's secret key 'S', random number 'r' and the ID and PW of U_a. The random number r, B is send to the user via a secure communication channel. The mobile app computes B' using Key_a and the received random number 'r', where Key_a is the master secret key stored in the mobile token within the phone.

UA2. Mobile app checks whether B' = Challenge B, received from AS. If so, mobile app considers the message as being received from an authenticated source. Mobile app sends $K = HMAC(Key_a, B')$ to IdP. IdP on receiving the message K, computes $K' = HMAC(Key_a, B)$ and checks whether it is equal to the received K. If equal IdP considers the user as authenticated and that the integrity of message is maintained.

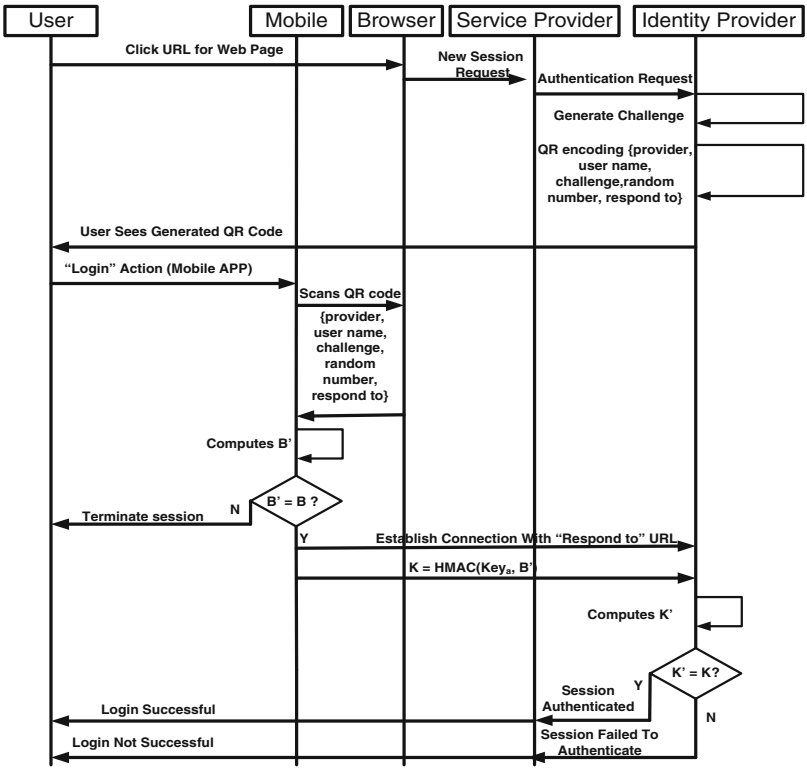

Fig. 7 Login and authentication phase of broker-based protocol using mobile phones

UA3. IdP creates an assertion token containing the result of the authentication process. IdP sends the token to the SP. The SP verifies the token issued by the IdP. It is assumed that the IdP and SP works in a trust-based environment. If the authentication is successful and the token is valid then SP notifies the user's browser of a successful login. Otherwise the login request is rejected.

Password Change Phase The password change phase is invoked when the user wishes to change his password without the intervention of the IdP or the SP is carried out as follows:

UP1. User enters his identity (ID_a) and Password (PW_a) and executes the 'Password Change' request. The mobile app computes $m' = Key_a \oplus h(ID_a \| PW_a)$ and checks whether it is equal to stored 'm'. If equal, the mobile app prompts the user to enter the new password '$PW_{a\ new}$'. Otherwise the 'password change' request is rejected.

UP2. The app calculates $h(ID_a \| h(s)) = Key_a \oplus h(PW_a)$. Then the app computes $Key_{a\ new} = h(PW_{a\ new}) \oplus Key_a \oplus h(PW_a)$ and $m_{new} = Key_{a\ new} \oplus h(ID_a \| PW_{a\ new})$ and replaces the existing values in the file with the new values.

6.3 Protocol for Direct Authentication Using Smart Card/Crypto Card/Audio Pass as the Second Factor

Phases of the Protocol The proposed scheme consists of four phases, viz., Initialization, Registration, Login and Mutual Authentication, and the Password change phase as explained in the following paragraphs. The notations used are listed in Table 3.

Initialization Phase RA selects two prime numbers p and q. RA computes $n = p * q$ where n is public and p and q are secrets. RA chooses a master secret key 'x' and a secret number 'd'. The keys of RA should be generated and managed using standard hardware security management (HSM) module.

Registration Phase In this scheme every server and user has to first register with RA. This phase is divided into two subphases: (1) Service provider integration with RA and (2) User registration phase.

Service Provider Integration with RA S_j selects its identity SID_j and submits the registration request to RA. RA after verifying S_j, computes SS_j and sends $\{SS_j, h(d), n\}$ to S_j via a secure channel. It is assumed that RA and S_j communicates using signed messages and thus works in a trusted environment.

User Registration Phase The registration process as shown in Fig. 8 can be explained as follows. U_i chooses his identity ID_i, a PIN number 'S' and submits $\{h(ID_i), S\}$to RA. RA generates a random number 'r' and computes the Initial secret of U_i as I_i. RA selects a random password PW_i for U_i and computes V_i, B_i, C_i, D_i, E_i, Y_i. RA stores $\{C_i, D_i, E_i, Y_i, n, h(.)\}$ into the smart card which is issued to U_i via a secure channel. RA sends PW_i to U_i via an SMS and updates the user table.

Login and Authentication Phase Login and authentication phase runs independently on each SP and is executed once for each session. When U_i wants to access the services of S_j, he carries out the login process, which can be explained as follows.
U_i requests for a protected resource of S_j by typing the URL or by clicking the link for logging in. The login page is displayed. User is prompted to swipe the smart card and click 'proceed' button. The rest of this phase will be executed on the basis

Table 3 Notations used in direct authentication using smart cards

RA, SP, SC	Registration authority, service provider, smart card
U_i, S_j, SID_j	ith user, jth service provider, ID of jth service provider
ID_i, PW_i, S	Unique identification of U_i, password of user U_i, pin number chosen by U_i
p, q, n, x, d	Prime numbers, public parameter, master secret, secret number of RA
I_i	Initial secret unique to U_i generated by RA
r, k, N_1, N_2	Random number generated by RA, SC respectively, nonce values
$h(.)$, $\oplus$, $\|\|$, $\Rightarrow$	Hash function, XOR, concatenation, secure communication channel

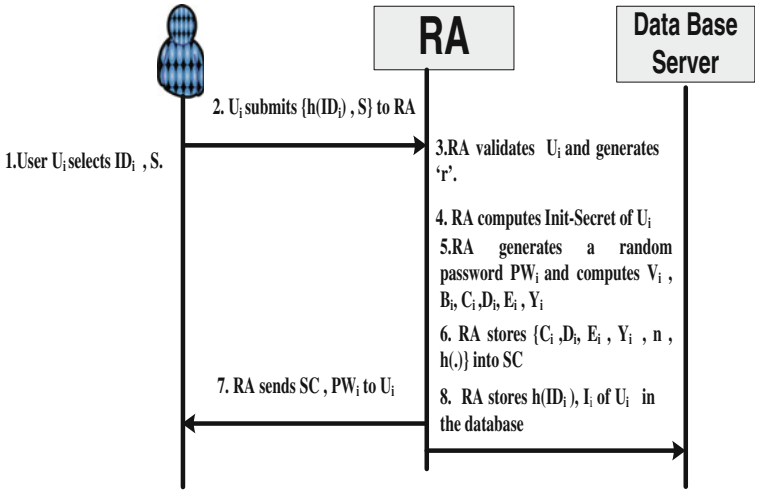

Fig. 8 Registration phase of direct authentication-based protocol using smart card

of the following three possible scenarios. **(i) User is not a registered user**: U_i has not registered with RA and hence does not have a smart card to swipe. In this case the user is directed to the registration page of RA and the registration phase is carried out **(ii) User has entered the smart card but is an invalid user**: U_i inserts his smart card into the reader. U_i is prompted to enter his ID_i and PW_i. SC computes V_i', I_i', D_i' and compares D_i' with D_i stored in the smart card. They are not equal and SC terminates the session. (**iii) User has entered the smart card and is a valid user:** U_i inserts his smart card into the reader. U_i is prompted to enter his ID_i and PW_i. SC computes V_i', I_i', D_i' and compares D_i' with D_i stored in the SC. If they are equal, the system checks whether U_i is accessing any SP for the first time. If so the system prompts U_i to change his password and the values in the smart card are modified accordingly. In this manner, the true password of the user is neither stored at the RA or nor does it travel across the network. Then SC proceeds to generate the login request message.

SC generates a random number 'k' and computes the parameters N_1, B_i, K_i, Q_j, P_{ij}, CID_i, Z_i. SC sends $\{h(\mathrm{ID}_i), P_{ij}, \mathrm{CID}_i, M_1, Z_i\}$. On receiving the request from U_i, server S_j checks whether there is an entry corresponding to $h(\mathrm{ID}_i)$ in its database. If it exists then the corresponding init-secret, I_i is retrieved from the local database. Otherwise a request for init-secret, is sent to RA along with the $h(\mathrm{ID}_i)$. RA retrieves the same from its database and sends init-secret I_i to S_j. S_j updates its user table with username and Init-Secret. Server S_j computes k and N_1. S_j checks the freshness of the nonce N_1 and computes Q_j', Y_i, E_i, B_i', D_i', P_{ij}, CID_i, M_1'. S_j compares M_1' with the M_1 in the login request. If the equality does not hold S_j rejects the request. Otherwise S_j generates a nonce N_2 and computes M_2, E_i', M_3. S_j sends $\{M_2, M_3\}$ to U_i. On receiving $\{M_2, M_3\}$, U_i computes N_2 and M_3*. U_i compares M_3* with M_3 in the response message from S_j. If equal U_i authenticates S_j successfully and

generates a mutual authentication message M_4 and sends $\{h(N_2 + 1), M_4\}$ to S_j. If M_3* is not equal to M_3, U_i terminates the session. On receiving $\{h(N_2 + 1), M_4\}$, S_j computes $h(N_2 + 1)$ and checks the freshness of the nonce. Then S_j computes M_4' and compares with the received M_4. If equal then mutual authentication holds. Otherwise S_j terminates the session. After mutual authentication, both U_i and S_j compute the session key $SKey_{ij}$.

7 Conclusion and Future Work

This work proposes a user authentication framework for Cloud environment, which attempts to address the authentication issues in Cloud. The authentication architecture and protocols of the proposed scheme addresses the issue of maintaining multiple authentication credentials in a multi-server environment by adopting SAML Single Sign-on functionality. The framework also caters to the requirements of those departments/organizations that prefer to independently authenticate their customers and this is made feasible by providing the option of executing the authentication protocol by the CSPs. The authentication protocols use two-factor authentication where password is the first factor and Smart Card/Crypto Card/Audio Pass/USB Token/Mobile Phone is the second factor. Security analysis of the protocols has been done and the protocols are resistant to user impersonation attack, server impersonation attack, replay attack, insider attack, parallel session attack, smart card lost attack, and the like.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: a berkely view of cloud computing. Technical report no. UCB/EECS-2009-28. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf
2. Amlan, J.C., Pradeep, K., Mangal, S., Hyota, E.L., Hoon-Jue-Lee: A strong user authentication framework for cloud computing. IEEE Asia-Pacific Services Computing Conference (2011)
3. Chakraborty, R., Ramireddy, S., Raghu, T.S., Rao, H.R.: The information assurance practices of cloud computing vendors. IT Prof. **12**, 29–37 (2010)
4. Miller, H.G., Veiga, J.: Cloud computing: will commodity services benefit users long term? IT Prof. **11**, 29–39 (2010)
5. Blumenthal, M.S.: Hide and seek in the cloud. IEEE Secur. Priv. **8**, 29–37 (2010)
6. Ponemon, P.L.: Security of cloud computing users. Ponemon institute, research report. http://www.ca.com/files/industryresearch/security-cloud-computing-users_235659.pdf (May 2010)
7. Gens, F.: New IDC IT cloud services survey: top benefits and challenges. IDC exchange. http://blogs.idc.com/ie/?p=730 (2009)
8. Almulla, S.A., Yeun, C.Y.: Cloud computing security management. II International Conference on Engineering Systems Management and its Applications (ICESMA) (2010)

9. Celesti, A., Tusa, F., Villari, M., Puliafito, A.: Security and cloud computing: inter cloud identity management infrastructure. 19th IEEE International workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010, pp. 263–265
10. Rui, J.: Advanced secure authentication framework for cloud computing. Int. J. Smart Sens. Intell. Syst. **6**(4), (2013)
11. Jeremy, K.: One of the most convincing phishing attacks yet tricks you with Dropbox sharing. PCWorld. http://www.pcworld.com/article/2835892/dropbox-used-for-convincing-phishing-attack.html (Oct, 20 2014)
12. Robert, M., Oops! amazon web services customer unleashes 'denial of money' attack—on himself. WIRED. http://www.wired.com/2012/04/aws-bill-in-minutes/ (April, 2012)
13. CRM Provider Salesforce Hit With Malware Attack: Entrust. http://www.entrust.com/crm-provider-salesforce-hit-malware-attack/ (September, 2014)
14. Darreni, P.: Google app engine has thirty flaws, says researcher. The register. http://www.theregister.co.uk/2014/12/09/google_app_engine_has_thirty_flaws_says_researcher/ (December 2014)
15. Jiang, R.: Advanced secure user authentication framework for cloud computing. Int. J. Smart Sens. Intell. Syst. **6**(4), (2013)
16. Nayak, S.K., Mohapatra, S., Majhi, B.: An improved mutual authentication framework for cloud computing. IJCA **52**(5), (2012)
17. Takabi, H., Joshi, J.B.D., Ahn, G.: SecureCloud: towards a comprehensive security framework for cloud computing environments. Proceedings of IEEE 34th Annual Computer Software and Application Conference Workshops, pp. 393–398, 19–23 July 2010
18. Falas, T., Kashani, H.: Two-dimensional bar-code decoding with camera-equipped mobile phones. Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 597–600, 19–23 March, 2007
19. OASIS: Security assertion mark up language, V2.0, Technical overview. http://docs.Oasis-open.org/Security/Saml/Post2.0/sstc-saml-tech-overview-2.0-cd-02.html