Secure Inter-Cloud Federated Identity Management using IID

Monika K. Katwe
M.E pursuing (Computer Engineering)
K.J Somaiya College of Engineering, Mumbai.

Manish M. Potey
Asoc. Prof. Department of Computer Engineering
K.J Somaiya College of Engineering, Mumbai.

ABSTRACT
The proposed system support single sign on in inter-cloud environment where user can manage in different cloud environments and provide single set of credential to access different Saas cloud application provided by different cloud service provider without re-authentication. Single sign on defines the ability to authenticate only once in a distributed network and to access several protected services and resources without re-authentication. To achieve this feature the system support federated identity management system. The federated identity management system crosses organizational boundaries. To manage identities of user in this case, a cooperative contract need to be set up between multiple identity providers, using a centralized approach. The proposed system uses third party auditor or third cloud to synchronize the identities of user among different clouds. As the user data are transferred or exchanged between different clouds environment the chances of stealing the data is increased. To avoid this the system is secure from some attacks like identity theft, denial of service etc. and also secure channel is maintained to transfer/exchange information between different clouds.

Keywords
SSO, Authorization, Authentication, Re-authentication, Cloud environment, IID, Electronic identification, Federated identity management, Identity federation.

1. INTRODUCTION
The growing area of cloud computing offers various features for user comfort like elasticity, scalability, multi-tenancy, pay as you go option so the user can pay according to their usage level and customization through which user can customize their required resources. Due to these features cloud computing demand is getting increases in various areas like e-government sector, e-health etc. According to the National Institute for Standards and Technology (NIST) cloud computing categorize into three different service delivery models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [1]. IaaS provides virtual computing resources such as virtual machines, networks and data storage, CPU cycles. PaaS offers programming run-time environments where customer can implement and run web application. In the third service model (SaaS), software applications are hosted by cloud service provider and provided as a service over the internet. To satisfy the current market need the cloud computing must employ in all IT areas where web application can easily deploy on cloud and provides software as a service to users. The transfer of such applications to the cloud has a couple of advantages, e.g. less maintenance efforts or lower costs [1]. But cloud service need to provide some level of security to such application. However, such security requirements are identification, authentication and secure system from attacks. The web applications in cloud environment are generally secured by using Username/password mechanism. But this is very weak technique to secure the whole system. The n factor authentication and identification mechanism must be used to secure the SaaS services in some sensitive areas like e- government sectors.

In current cloud authentication, user has to authenticate separately to access different SaaS application for that user has to enter or managed different set of credential to access various application which may increase time for authentication and user frustration by remembering set of log in credential. Cloud service provider also has to maintain different database for identifying user for each application which increases the cost of system. To avoid this system must support SSO (SSO) mechanism in same as well as different cloud environment.

To overcome above drawback the given approach demonstrated the two factor authentication and use of IID[1] solutions to achieve SSO for secure cloud authentication in inter-cloud environment. In two factor authentication scheme, firstly user has to enter their log in credential like log in-id, password, pattern and in second level user can identified through IP filtering and One time password (otp) scheme. Furthermore, SSO can achieve by using IID interoperability framework in inter-cloud environment. By using SSO the usability and user comfort can increase. Through the SSO the users are seamlessly authenticated to several inter-cloud services by their IID only once for authentication. The proposed system demonstrated SSO authentication between two public cloud service providers. The use of IIDs for cloud authentication paves the way for increasing future cloud adoption in sensitive areas such as e-Government or e-Health, as legal requirements can be easier fulfilled compared to username/password authentications. Proposed system uses the two level authentication schemes to access the local application where the user has registered over weak username/password authentications and the concept of IID is used for authenticating and identifying the user in inter-cloud environment, accordingly SSO can achieve. Through the IID solution user can able to access the applications provided by other cloud service provider where the user has not registered.

For achieving SSO in inter-cloud environment, the concept of federated identity management must be required where credential or identities of all the users manages to their respective cloud which avoids remembering and entering different credential for different application as well as must be manage in third party auditor through which user can be known to other cloud service provider. The proposed approach also provides federated Identity management which uniquely identifies the user in inter-cloud environment. It manages the identity of all the users associated with different cloud in third party auditor which provides centralized authentication, where IID's of the entire user getting stored

and managed. The system also tries to handle some type of attack like denial of service, identity theft, reply attack etc. So that user privacy can be maintained from attacker and user can securely access the n applications provided in inter-cloud environment. Furthermore, the system also uses AES encryption to securely upload and download of file so; a secure channel is maintained, to achieve privacy of the user and integrity of the data.

The remainder of the paper is organized as follows. Section 2 describes the concept of SSO and some protocol as a literature survey related to proposed system. The inter-cloud SSO identity management architecture using IID is presented in Section 3 and some implemented sample snapshot is described in section 4. The section 5 explains some threats avoided by the system. Finally, draws conclusions in Section 6.

2. LITERATURE STUDIED

This section gives some background study of current model. Here, the concept of SSO, existing identification and authentication approaches and some protocol relating to given work are briefly discussed.

2.1 Single Sign-On (SSO)

User access the set of Saas application after crossing authentication and identification mechanism. These applications are provided by cloud service provider and provider validate the user by running separate user management for each application. That means user has to remember and enter the set of log in credential for accessing a separate application provided by same or different service provider. This repeated authentication step lowers the user comfort. To overcome such drawback, the concept of SSO (SSO) came into existence. The SSO mechanism allow the user to access multiple application by performing authentication process only once which avoids frequent re-authentication so the user reliability can increase[2]. In SSO mechanism user has to authenticate only once for accessing application with one cloud service provider which results automatically authenticate with other cloud service provider. Using SSO user has to remember single set of log in credential which avoids remembering a bunch of log in credential and remove burden on user resulting saves authentication time and cost. Due to this a single user management needs to run at cloud service provider side which improves security and avoids maintenance of multiple databases. If someone steals your credential then complete SSO system is available to the attacker and If the user management fails then user can not able to access single application[2]. The whole system is disturbed these are the biggest disadvantage of SSO system. The given system handles some attack which provides some level of security and tries to avoid this drawback in some extent.

2.2 Protocols

Following are standard protocol supported in inter-cloud environment and has been widely used for implementation of SSO system.

1. **Security Assertion Markup Language (SAML):** The SAML [2] is based on XML open standard format. It is basically designed for the exchanging of authentication and authorization data in secure form between different providers. SAML does not specify the method of authentication; it can be username/password or multifactor authentication. It can be applicable to Web SSO; Attribute- based authorization, Securing web services.
2. **WS-Federation:** WS-Federation [2], is designed for enabling identity federation across different security realms and it is based on XML-based specification. It is a part of the WS-Security framework. Microsoft's Windows Azure cloud platform relies on WS Federation. It Uses security token services (STS) for authentication.
3. **OpenID:** OpenID is a decentralized SSO approach especially used for web-based services. In OpenID Users typically authenticate by username/password authentication mechanisms and receive a URL-based OpenlD identifier. Authentication is managed by OpenID providers. OpenID allows user to use an existing account to sign in to multiple websites, without needing to create new passwords. Google for example is an OpenID provider.
4. **Shibboleth:** In this protocol the federation process between the identity providers is conducted on the basis of the list containing the names of the providers and some predefined rules. This common rules act like agreement used between the providers. One of the lagging features of this model is the complexity of the management of the provider's list. It Uses user organization Log in credential as authentication mechanism and send minimum details to service providers
5. **OAuth:** It is standard open protocol focusing on application authorization. The Oauth supports an API which enables the one application to access specific user data of another application[2]. It is a popular in social networks such as Facebook, Linkedln or Twitter. It Uses limited access OAuth Token i.e. valet key as authentication mechanism.

The proposed system uses the concept of OpenID for generation of IID; by using this secure access of cloud application in inter-cloud environment is possible.

2.3 Need of Inter-cloud Identity Management System

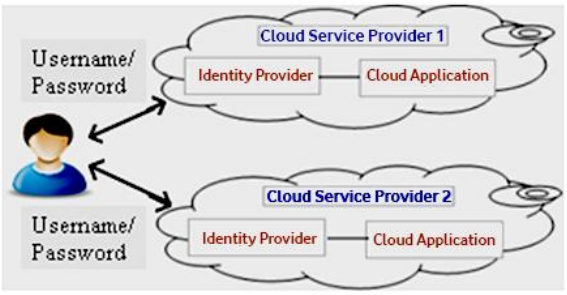

Figure 1: Current situation for cloud authentication [2].

In today's world every cloud service provider offer various SaaS application for satisfying user need and these applications are generally secured by username /password authentication mechanism. Each cloud service provider hosts its own and separate user management for managing users credential. This type of mechanism has some drawback like username/password is very weak authentication scheme in some sensitive areas like Government sector and another drawback is, user have to remember different set of credential for accessing cloud application. This repeated amount of different username/passwords decreases the level of security, because user has a tendency to re-use them or write them close to their computers. That means, the user first has to register at each provider and second has to authenticate

separately. Figure 1 illustrates this current situation for cloud authentication.

It shows that user tries to access the two different applications running on different domain. Each cloud service provider runs its own user management for each application in its domain. If user want to access application provided by cloud service provider 1(csp1) then user has to registered with identity provider1(ID1) of csp1 simultaneously for accessing application 2 user has to registered separately with IDP2 of csp2. After successful both registration user need to authenticate separately for each application by entering their respective registered log in credential. That means user has to provide separate set of registered credential to respective identity providers for authentication. In this situation it creates burden on user to remember separate set of credential and authenticate separately. To overcome this drawback, the proposed system introduces an inter-cloud SSO authentication. By applying this concept, users need to authenticate only once but still get access to applications of multiple cloud service providers. To implement secure SSO in inter-cloud environment, users normally just need to register with any one of IDP and need to remember one set of credential. So the risk of writing down near to computer or diary gets reduced. Another option for increasing authentication security is the use of IID. IIDs allow for unique user identification and strong user authentication. For instance, the proposed design supports concept of IIDs which can be renewed after some threshold for security reasons. The proposed architecture combines both, secure authentication using IIDs and SSO to take advantage of the benefits of both solutions for an inter-cloud SSO authentication scenario.

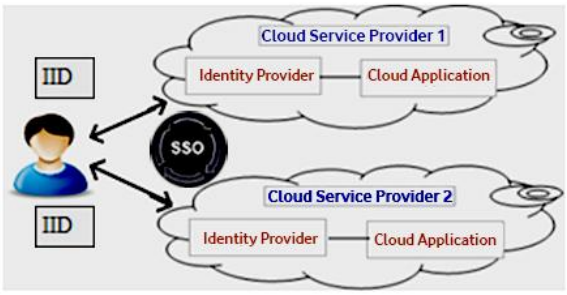

Figure 2: Architecture for inter-cloud SSO [2].

The inter-cloud architecture shown in Figure 2 supports strong IID authentication at different SaaS cloud service application, providing SSO[3] between those applications at the same time. This means, by using system generated IID a user needs to authenticate once at one cloud service application. After that, the user is automatically and seamlessly authenticated for other cloud application.

3. IMPLEMENTATION OF IDENTITY MANAGEMENT SYSTEM IN INTER-CLOUD SSO

3.1 Terminologies

1. **Identity provider (IDP):** identity management system must address the provisioning and de-provisioning of user identities within an organization. IDP [4] [5] is used to manage the identities (Basic information) of user which were registered for accessing cloud application. It focuses on the authentication of the users as well as on the management of identity information, which can be shared with IDP of other cloud.
2. **Third party auditor:** The third party auditor or third cloud is responsible to collect as well as synchronize all user identities associated with different clouds. All the identities of users associated with other cloud are centrally managed in third party auditor. Based on the gathered information third party auditor permits the user to redirect other SaaS cloud application.
3. **Inter-cloud ID (IID):** It is unique number generated by the system for each user, which uniquely identifies the user from other users in inter-cloud environment. IID is centrally managed in third party auditor as well as managed in respective IDP. It can be renewed after some threshold value for achieving security.

3.2 Proposed Design

The Proposed system adopted inter-cloud framework to support authentication at different cloud services on the one side, and to support SSO between these different applications on the other side. This model consider three different clouds, where following applications are deployed on two clouds to achieve inter-cloud environment and third cloud is used as third party auditor i.e. cloud C which centrally manages the identities of all the users associated with different clouds. Based on stored identities, third party auditor gives the permission to users for accessing SaaS application of other cloud using IID only. That means in third party auditor all the identities of IDP1 and IDP2 is centrally managed and the users are registered with IDP1 or IDP2 to access SRP and ME and RU application. For example If user is registered with IDP1 in cloud A to access SRP and ME application and want to access the application of cloud B then third party auditor can give the permission to access other cloud application by validating their IID, which is managed in third party auditor. Users don't have to enter their credential again. Accordingly, it can achieve SSO in inter-cloud environment by managing the identities in third party auditor. The model presented in figure 3 shows proposed design.

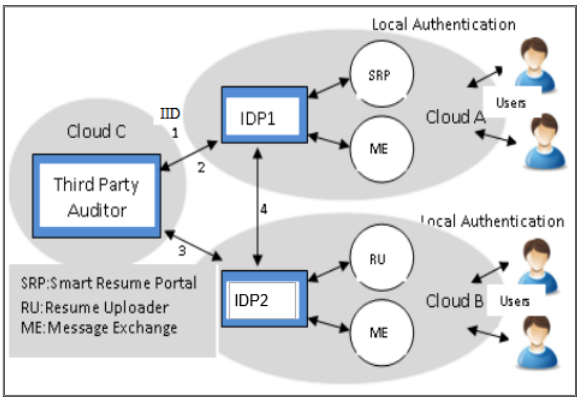

Figure 3: Proposed system model.

Following are the applications deployed on clouds to show SSO in inter-cloud environment.

1. **Smart Resume Portal(SRP):** SRP application hosted by cloud A and is used for filtering of resume. All the sales person are registered with IDP1 and can able to access the resume of engineer person through SRP application depending on their project requirement.
2. **Resume Uploader(RU):** RU application is hosted in cloud B. It is used to upload the resume by the engineer person. Depends of their skill and experience sales person can select their resume for further task.

3. **Message Exchange (ME):** ME application hosts in both cloud A and B. It is used to perform the inter-cloud communication between sales person and engineer person over secured lines. Both of users are registered with different clouds.

3.3 Authentication steps

The proposed system provides trust based mechanism between different cloud providers and third party auditor. This system performs different levels of authentication for local SaaS application and application belongs to other cloud.

Algorithm for authenticating the user for accessing application running on same cloud

Security and Provisioning are primary concern for cloud providers in cloud environment. The current System uses a secure cloud authentication for accessing Saas application provides SSO. The identities of cloud A and B user are managed by IDP1 and IDP2 respectively. Here two level of authentication is used, in first level user have to enter their log in credential and in second level, Perform IP filtering or one time password method to securely authenticate the user. Following are the set of steps which is used for authenticating the users for accessing SaaS application of their registered cloud. By using this algorithm SSO can be achieved between applications running on same cloud.

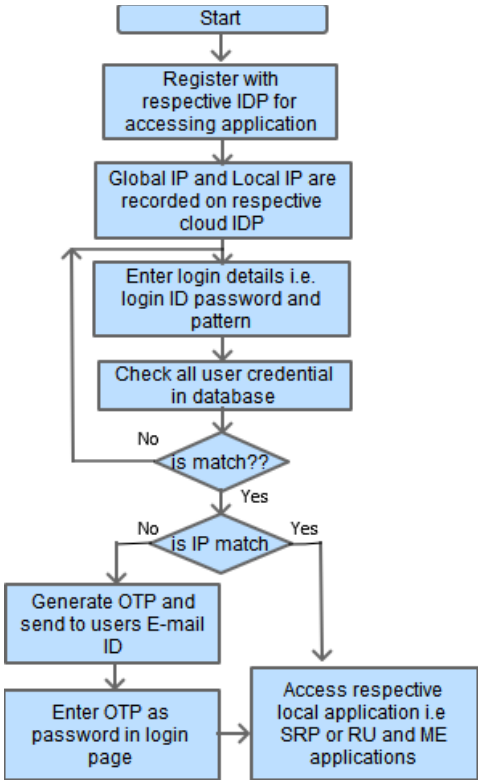

Figure 4: Data flow diagram for local application authentication on Cloud A and B.

1. An application programming interface is displayed where option of creating and managing user account for cloud application is mentioned. User have freedom to register, access update their credential on cloud portal. When the user registers first time, his/her IP address is recorded by the system automatically.

2. When user want to log in for application the respective log in page is displayed on user screen and user need to click on log in option to access particular application like SRP, RU etc.

3. This system uses log in id, password and pattern as an authentication criteria in first stage. When user enters their credential on log in page, the credential as well as IP is verified and site will redirect to the user interface. Now user has a right to access the application. If user is accessing from other machine definitely IP is not matched, then one time password is automatically generated by the system and send it to his/her email-id as a second stage authentication criteria. If IP is not matched then generated otp need to enter as a password field in log in page. The otp must be submitted within specified 15 min thresholds then only system will change previously recorded IP and set current machine's IP in the database and system will redirect to respective application otherwise user has to generate another otp. otp can also generate for avoiding identity theft attack.

The above steps are mentioned in figure 4 and can be performed i.e local authentication, when user tries to access the application of cloud where user has registered. But when user tries to access the application of other cloud then user don't have to enter their credential again. User can get access by entering IID only. The system uses concept of OpenId[6] for generation of IID

Algorithm for authenticating the user for accessing application running on other cloud

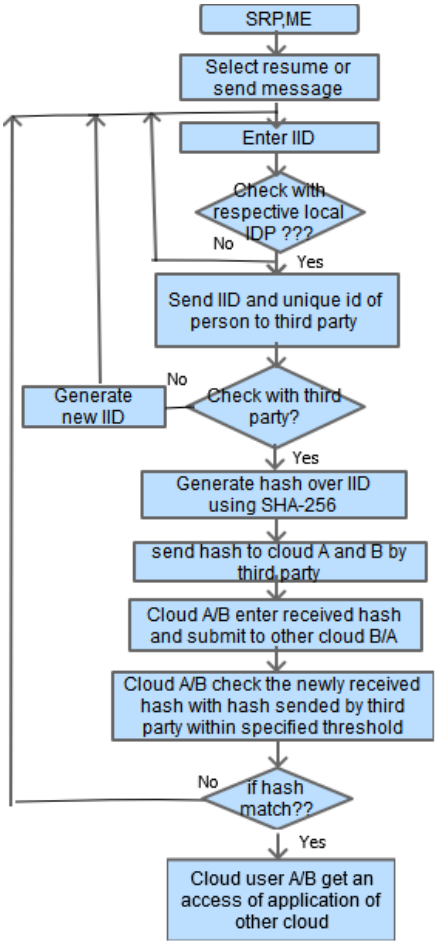

Figure 5: Data flow diagram for non local application authentication on Cloud A and B.

By using federated identity management in inter-cloud environment, user can able to access the application of other cloud from their registered cloud without re-authentication. Following are the set of steps mentioned in figure 5 for accessing Saas application running on different cloud. It is assumed that cloud A user want access the application of cloud B. by using following steps SSO can achieve in inter-cloud environment using IID. The system generates Inter-cloud ID (IID) for accessing application running on other cloud. It is automatically generated by the system when the user register first time, and it can be renewed after some decided threshold. The system provides 30 days threshold for renewal of IID to avoid some attacks and increased security. There is also provision inside system to generate new IID wherever user wants.

4. When the user wants to redirect to some other cloud application, he/she has to enter recent IID for uniquely identifying the user in inter-cloud environment.

5. The IID is currently verified with local database i.e cloud table maintained locally on respective cloud where user is registered and then it will encrypt using AES[7] algorithm. Encrypted IID will send to third party auditor shown in step1 in proposed model.

6. In third party auditor decryption is performed and verifies IID with user identities managed in database.

7. If IID matches then one token is generated over IID using SHA-256 algorithm and encrypt the token using AES algorithm and send it to Cloud A and B both. Step 2 and 3 indicates sending of token to cloud A and B

Received hash token gets decrypted at both clouds. Cloud B stores received token into its database along with user identities (like who is going to access the application) and starts the timer for some threshold value. Here provides only 10 min thresholds to verify whether the third party auditor gives the permission to cloud A user or not.

8. On the other side one auto fill token page will appear on cloud A user's screen. User has to submit the token within specified threshold i.e. 10 min for getting access. If the user is busy somewhere and not able to submit then he has to generate another token, if user has successfully submitted the token within specified threshold then encrypted token will send to cloud B mentioned in step 4.

9. On cloud B side decryption of token is performed. Now the cloud B will have two token, one is received from third party auditor and other is from cloud A, if both token matches then cloud A user successfully redirect to cloud B to access the SaaS application. It indicates that third party auditor gives a permission to cloud A user to access the application of cloud B. Vice verse is also possible. To achieve the security, some threshold is applied to submit the token.

The given system maintains the database for managing the identities of user in each cloud for audit of the user (user operations) so that the accountability also can be shown. Some database table maintained at cloud A, B and third party auditor is mentioned in figure 6. When the user has successfully redirected, the secure channel is maintained between user and cloud application as well as between two different cloud applications associated with two different clouds. It uses AES 128 bit encryption for maintaining secure channel. So further communication can performed through secure channel only, accordingly integrity can be achieved.

3.4 Example

Consider the example, Cloud A user i.e. sales person want to access resume of engineer person. The resumes are uploaded by engineer person associated with cloud B through Resume upload (RU) application. That means resumes are stored at cloud B and sales person associated with cloud A access resume from Cloud A using Smart resume portal application (SRP). As mentioned earlier in proposed model SRP and ME are deployed at cloud A and RU, ME are deployed at cloud B. figure 6 shows the mapping relation using database table implemented in system, which indicates that how the user is redirected from cloud A to cloud B to access resume from one application to another using IID concept. Same procedure is followed to perform inter-cloud message communication.

4. SAMPLE SNAPSHOT IMPLEMENTED IN SYSTEM

4.1 SRP

By using SRP application registered sales person can able to access the resume of engineer person based on their requirements. The entire sales person must be approved by the identity provider then only sales person can able to access the resume. Based on their role i.e. designation they can access the engineers resume that means not all the sales person have access of all engineers resume here system provides role base access for accessing resume. Sales user can also search resume by selecting multiple technologies, required skill and experience and accordingly message can send to respective person. Furthermore, sales user can also see how much resume is selected and the team formed, accordingly they can add or delete the resume from formed team. The sales person can view or download the resume. Downloading the single resume as well as downloading all resume of formed team is possible. The sample snapshot of SRP application shown in figure 7.

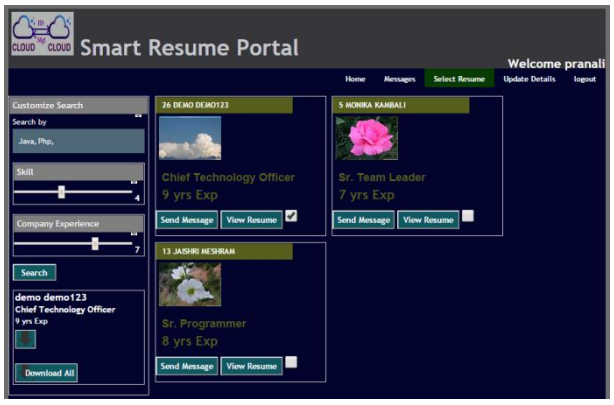

Figure 7: Sample snapshot of SRP application for resume selection.

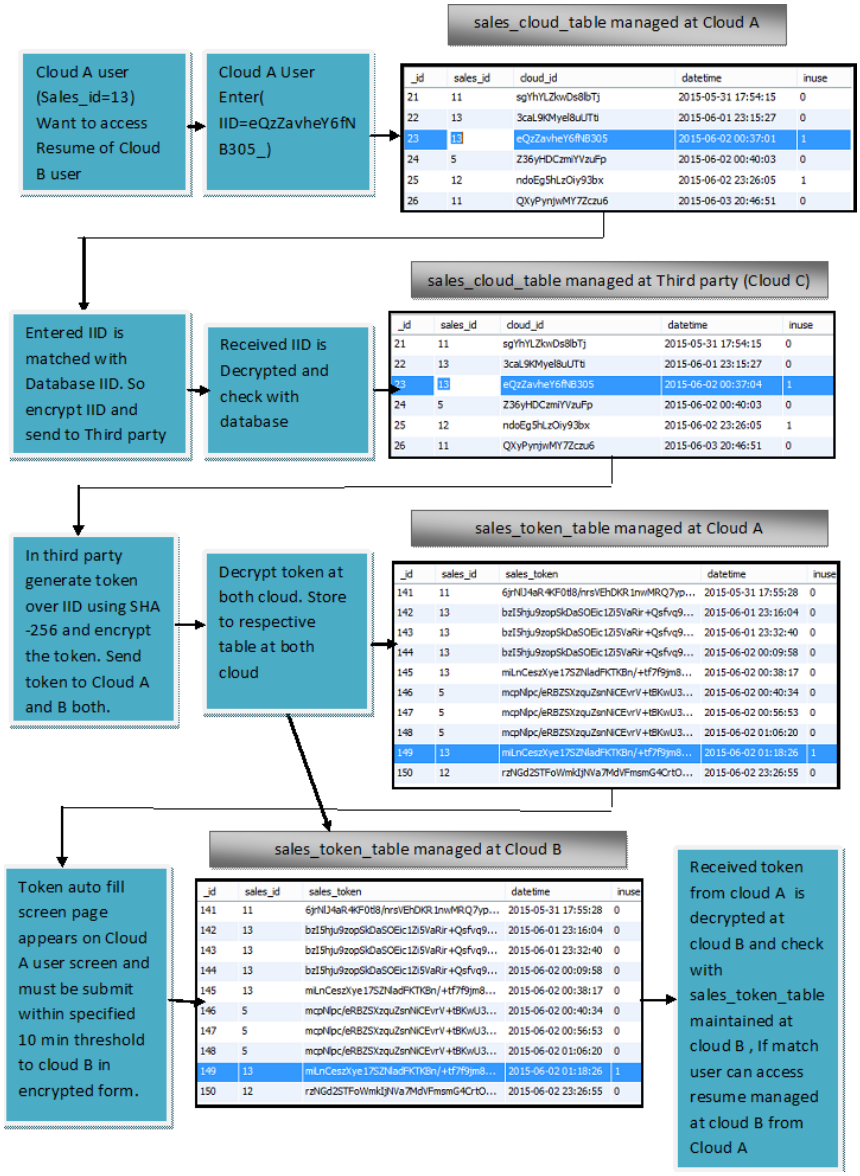

Figure 6: Mapping relation between database tables.

4.2 RU

Figure 8 shows the snapshot of resume upload application where engineer person can upload their resume to cloud in encrypted format using AES encryption and 128 bit key is also generated by the system automatically for uploading resume. The generated key is displayed by using asterisk form to achieve security. The resume upload table is also maintained at cloud B for keeping a track of resume upload by engineer person. In resume upload table the key as well as file path is maintained in encrypted format along with file extension and date time of upload. The actual file is stored in build folder of RU application.

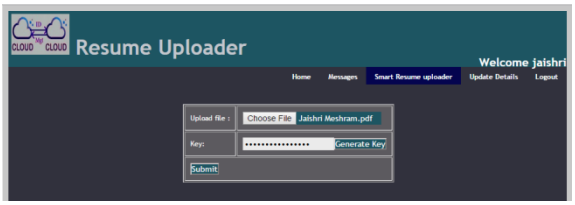

Figure 8: User interface for uploading resume.

4.3 ME

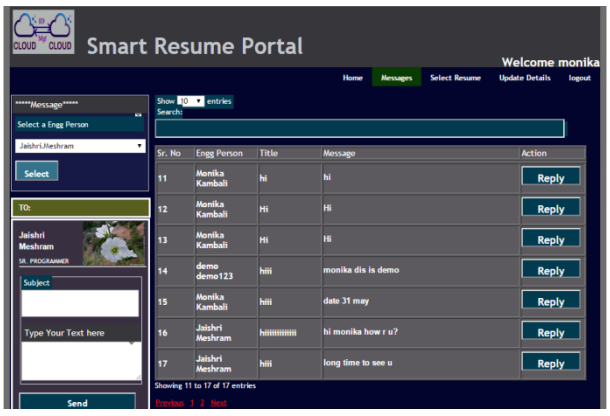

Figure 9: Message Exchange user interface of Cloud A.

Message exchange application deployed on two different clouds separately to show inter-cloud communication. Figure 9 and 10 indicate the user interface for exchanging the message in encrypted format to achieve security. To implement inter-cloud communication the system maintains message table to their respective cloud. This shows the communication history between sales person and engineer person. Here message is encrypted using AES encryption and stored on database. Table also maintains readonly field indicates whether message is read by respective person or not. By using following user interface the respective person can perform send message operation by selecting person from list and can perform reply operation.

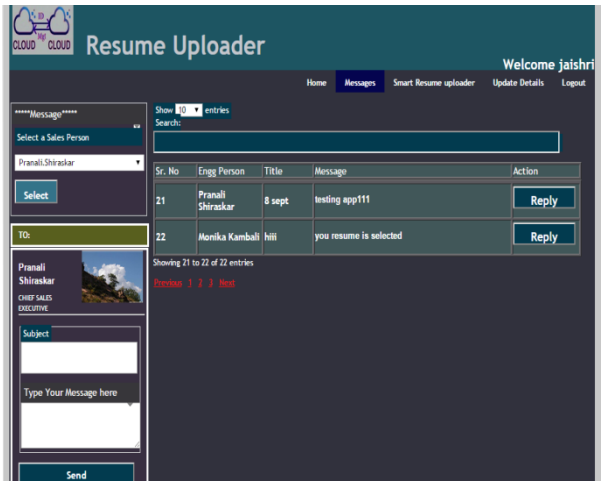

Figure 10: Message Exchange user interface of Cloud B.

5. SECURITY THREATS AND AVOIDANCE

The current system can be secure from following attacks.

1. **Reply attack:** In this type of attack, the attacker replays the previously stored result and presents himself as authorized user. To avoid replay attack system generates nonce before submitting any form and store to the database, After successful submission of form the current value of nonce will verify with the database value. If both value matches then indicates that the authorized user is submitting form for their required operation if not definitely nonce will not match and unauthorized user will not able to access application.

2. **Identity theft[8]:** If someone breaks computer security and steals private information example id, password etc by exploiting the natural tendency of a person or the user is suffered from social engineering or identity theft and attacker tries to access their application on behalf of victim user then system generated email will be send to victims registered email id. The current system uses location as a parameter to find out identity theft attack. If attacker access the victims account, definitely he/she will access from different location. Priory system records the registered user location into the database and that location will verify every time when user perform log in operation. If attacker steals all information and tries to access the account from different location then system will generate OTP and e-mail will send to victim's email-id indicates account is accessing from other location.

3. **Denial of service:** Dos attack can occur when authorized user are not able to access the application because attacker is continuously hits the application page. This can be avoid by no. of hitting request in a given threshold timestamp from same IP. If it exceeds the limit the automatic logout can be performed, here system uses 20 min timestamp to count no. of hitting request from the same IP address. Firstly we check whether currently executing file is JSP or not if it is jsp file and hitting request not exceeded the limit then that request is store on the database table. The system performs automatic logout when the no. of hitting request is exceeded the limit in last 20 min from the same IP address.

4. **Man In middle attack:** In this type of attack, the attacker placed himself in the middle of the channel and tries to get/access the user data. In this system a secure channel is built using AES encryption so, man in middle attack possible to eliminate.

5. **Non repudiation:** Non repudiation is a mechanism which guarantees that the message sender must not later deny having sent message and that the recipient must not deny having received message. In this system the user stored data on cloud so that the respective authorized user can only get the data by decrypting it on other cloud. Accordingly, data access is not possible by unauthorized user.

6. CONCLUSION

The system support federated identity management to achieve SSO in inter-cloud environment where user identities are managed in different cloud environments and can able to access various Saas application without re-authentication. The system provides more secure environment by adding two level authentications where in first level system uses user log in credential and in second level uses IP filtering and one time password as authentication criteria to access the local SaaS application where the user has registered. Inter-cloud SSO can be achieved by using the concept of IID. IID allows the user to access the application of other cloud where user has not registered, for that system manages the identities of all the users to their registered cloud as well as to third cloud, who give the permission to access the application of other cloud. As the information is shared between clouds, the system maintains secure channel between them and also tries to avoid various attacks. In future the system model can be extended by applying more secure authentication over given authentication scheme and current system supports known circle of trust where IDPs and third cloud known to each other. Furthermore, the system can be extended by incorporating unknown circle of trust.

7. REFERENCES

[1] "Federated Identity Management in Cross-Cloud Environment" by Monika K. Katwe, Manish Potey, International Journal of Advanced Computing And Electronics Technology (IJACET) of Volume-2, Issue-3, 5th may 2015. http://troindia.in/journal/ijacet/Vol2_Iss3.html

[2] "Secure Inter-cloud SSO (SSO) using IIDs" by Bernd wattendorfer, Arne Tauber E-Government Innovation Center (EGlZ) Graz University of Technology Graz, Austria.

[3] "SSO For Cloud" by Pratap Murukutla National Institute of Technology,Karnataka, K.C. Shet National Institute of Technology,Karnataka.

[4] "Identity management based security architecture of cloud computing on multi-agent systems" by R.M. Lguliev Institute of Information Technology ANAS Baku, Azerbaijan, F.C. Abdullayeva Institute of Information Technology ANAS Baku, Azerbaijan.

[5] Balasubramaniam, S.; Lewis, G.A.; Morris, E.; Simanta, S.; Smith, D.B., "Identity management and its impact on federation in a system-of-systems context," Systems Conference, 2009 3rd Annual IEEE , vol., no., pp.179,182, 23-26 March 2009 doi: 10.1109/SYSTEMS.2009.4815794

[6] Khan, R.H.; Ylitalo, J.; Ahmed, A.S., "OpenID authentication as a service in OpenStack," Information Assurance and Security (IAS), 2011 7th International Conference on ,vol., no., pp.372, 377 ,5-8Dec. 2011 doi: 10.1109/ISIAS.2011.6122782.

[7] Fatemi Moghaddam, F.; Karimi, O.; Hajivali, M., "Applying a SSO algorithm based on cloud computing concepts for SaaS applications," Communications (MICC), 2013 IEEE Malaysia International Conference on , vol., no., pp.335,339, 26-28 Nov. 2013 doi: 10.1109/MICC.2013.6805850

[8] Ghazizadeh, E.; Zamani, M.; Ab Manan, J.-L.; Khaleghparast, R.; Taherian, A., "A trust based model for federated identity architecture to mitigate identity theft," Internet Technology And Secured Transactions, 2012 International Conference for , vol., no., pp.376,381, 10-12 Dec. 2012

[9] Dreo, G.; Golling, M.; Hommel, W.; Tietze, F., "ICEMAN: An architecture for secure federated inter-cloud identity management," Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on , vol., no., pp.1207,1210, 27-31 May 2013.