A Review on Single Sign on Enabling Technologies and Protocols

Tayibia Bazaz
Department of Computer
Science & Engineering
Jamia Hamdard
New Delhi, India

Aqeel Khalique
Department of Computer
Science & Engineering
Jamia Hamdard
New Delhi, India

ABSTRACT

In today's digital era, users are increasingly accessing countless number of applications every day. For accessing these services, the users first have to authenticate themselves and need to maintain a separate set of username and password for each application. This led to the development of Single Sign On (SSO). This paper presents review on SSO enabling technologies and discusses SSO architectures, protocols and analysis related to growing use of SSO.

General Terms

User Authentication, Service Authentication, Single Sign On

Keywords

Single Sign On (SSO), Authentication, Multi-factor Authentication (MFA), SAML, OpenID

1. INTRODUCTION

In the present era of Internet, Application Service Provider (ASP) provides a standard interface to a countless number of users and also a standard connection point to various application providers. As almost each application has its own authentication mechanism, users need to go through multiple login steps. The user information and security are not correlated making the user management complicated and unsafe. In order to address the issues related to the user convenience and security, the commonly used technique is Single Sign On (SSO). SSO is an access control method which asks a user to login once and without any further login criteria, he/she is allowed to access the resources of multiple software systems securely. SSO helps in the integration of the security policy and user information [1].

Prior to SSO, a user was supposed to login with a new account each time a new application was opened. Hence, was supposed to memorize numerous passwords which is really a difficult task to perform. To deal with this, users usually preferred to go for simple and almost same passwords. This approach is easy but has a potential threat. Choosing simple passwords made a cracker's job easy. An attacker can guess the password and gain access to all of the confidential information. With the introduction of SSO, users are being freed from this menace. They just need to authenticate themselves once and then can easily access the multiple applications running on various domains securely.

The structure of this paper is outlined as follows; in Section 2, this paper discusses about the architecture, trust models and variants of SSO. In Section 3, discussion about SSO enabling technologies and protocols is done. Section 4 discusses the benefits and drawbacks of using SSO. This paper also discusses the combination of SSO with Multi Factor Authentication (MFA) in Section 5. In Section 6, some focus is laid on the thought that whether SSO is a blessing or a risk factor and Section 7 concludes the paper.

2. SINGLE SIGN ON (SSO)

2.1 Overview

SSO is an access control method which asks a user to login once and allows them to access multiple resources and services after successful login without being prompted to login again. Thus, SSO approach enables users to authenticate only once and then enjoy easy access to other applications securely.

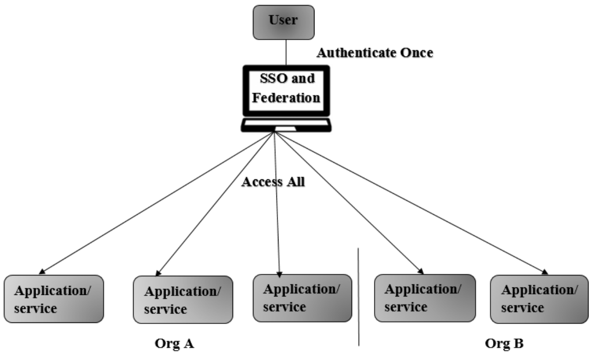

Fig 1: SSO Overview

Figure 1 shows the SSO approach where a user authenticates once and then can access different applications or services easily. These applications can be within a single organization or different organizations i.e., some are within one domain and some within multiple domains. Federation allows to access applications of different organizations and hence, takes SSO to the next level where users are able to federate their SSO solution outside their organization and allow trusted third parties to login once and use their applications [2]. SSO solution copies the necessary user credentials required across these domains securely.

2.2 SSO Architecture

There are different types of SSO architectures, with different properties and infrastructures namely Secure Client-Side Credential Caching, Secure Server-Side Credential Caching, SSO with Single Set of Credentials, Public Key Infrastructure based SSO, Token based SSO. Secure Client-Side Credential Caching and Secure Server-Side Credential Caching come under SSO with multiple set of credentials while Public Key Infrastructure based SSO and Token based SSO come under SSO with single set of credentials. Depending on their properties and usage, these architectures can be applied to various situations accordingly. The detailed description of these architectures is discussed below [3].

1) **Secure Client-side Credential Caching:** It is client based SSO solution. Here all the authentication related information is kept into a client-side credential storage. It allows the user to authenticate himself/herself once and afterwards the rest of the information for subsequent requests is being provided by the system automatically without the user's intervention. If the credentials provided by the end-users are valid then the users will be transparently authenticated to the other application servers. A high secure credential cache resides on client-side as shown in Figure 2. The cached credentials have to be stored securely as this cached information may be used to access some sensitive information or confidential web service. Thus, it's not advisable to be used from a portable client device or on some operating system having security issues. This solution has little flexibility as all the credentials are stored on a client-side credential cache. A user may face some sign-on problems while signing on via some other workstation. Also, the client-side credential cache has to be updated with the information of every new application server added.

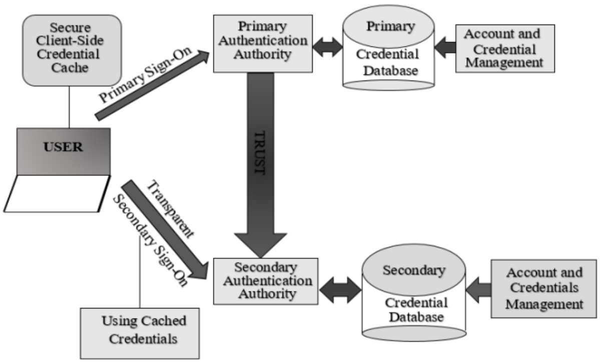
Secure Client-Side Credential Cache
Primary Sign-On
Primary Authentication Authority
Primary Credential Database
Account and Credential Management
USER
TRUST
Transparent Secondary Sign-On
Secondary Authentication Authority
Secondary Credential Database
Account and Credentials Management
Using Cached Credentials

Fig 2: Secure Client-Side Credential Caching

2) **Secure Server-side Credential Caching:** Also known as server based SSO solution where all the authentication details are stored in a central repository but the cache is stored on server side. The task of administering all the different passwords and providing the needed information directly to the application asking for them is done by the central server. Figure 3 shows Secure Server-side Credential Caching with two credential databases. These are primary credential database and secondary credential database. The primary credential database contains the primary credentials of different users and the mapping between primary credentials and secondary credentials while the secondary credential database contains only an image of secondary credentials. The mappings between these credential databases needs to be synchronized. The synchronization can be achieved in three ways. These are:

1) Integration of the credential synchronization services into the primary credential database.

2) Using an external software to handle the credential synchronization process.

3) Synchronization performed by the administrators themselves.

A trust relationship, depending on the need of credential synchronization, has to be set between the secondary authentication authorities and primary authentication authorities.

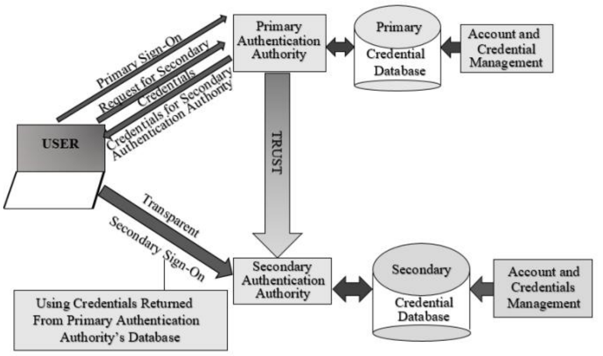
Primary Sign-On
Request for Secondary Credentials
Credentials for Secondary Authentication Authority
Primary Authentication Authority
Primary Credential Database
Account and Credential Management
USER
TRUST
Transparent Secondary Sign-On
Secondary Authentication Authority
Secondary Credential Database
Account and Credentials Management
Using Credentials Returned From Primary Authentication Authority's Database

Fig 3: Secure Server-Side Crendential Caching

3) **Single Sign-On with Single Set of Credentials:** The services that provide the management of services, implement the Single Sign On with single set of credentials. The feature of this SSO architecture is that it is well suited for homogenous environment where single naming account format and same authentication protocols are supported and identified by every entity in the whole network system.

4) **PKI-based Single Sign-On:** This approach makes use of the public key cryptography for user authentication. The system relies on the role of Certification Authority (CA) for the issuance and management of the digital certificates and hence users' digital identities. The user first has to identify herself/himself to an authentication authority which issues a public key certificate to the authenticated user as shown in Figure 4. Whenever the authenticated user wants to access a protected resource in subsequent authentication request, he/she creates a token and includes its digital certificate (public key) in it and signs it with her/his private key. On the reception of the request, the target server contacts the CA in order to verify the identity of the requesting user. There is a relationship of trust between the primary CA and the secondary CA as the latter's certificate is being issued by the former one. This enables any secondary CA to accept the certificate issued by the primary CA. The private key is a long series of random binary data and is hard to be postulated down on paper or to memorize but the key can be easily transmitted over a network, hence prone to thefts by any intruder. Examples of PKI based SSO solutions are Verisign, Globalsign etc.

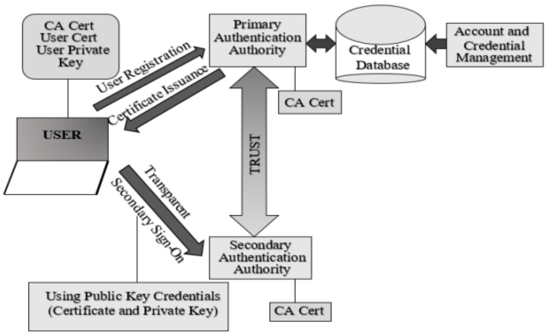
CA Cert
User Cert
User Private Key
User Registration
Certificate Issuance
Primary Authentication Authority
Credential Database
Account and Credential Management
CA Cert
USER
TRUST
Transparent Secondary Sign-On
Secondary Authentication Authority
Using Public Key Credentials (Certificate and Private Key)
CA Cert

Fig 4: PKI Based SSO

5) **Token-based Single Sign-On:** In this architecture, a user after signing into the primary authentication authority receives a temporary token as shown in Figure 5 which it can use further to access the resources or services without any re-authentication process. This is

possible as there is a relationship of trust between primary and secondary authentication authorities. Figure 5 shows that a user uses his temporary token to access the resource without being prompted to authenticate himself again to the Secondary Authentication Authority. An example for this authentication strategy is the Kerberos authentication protocol.

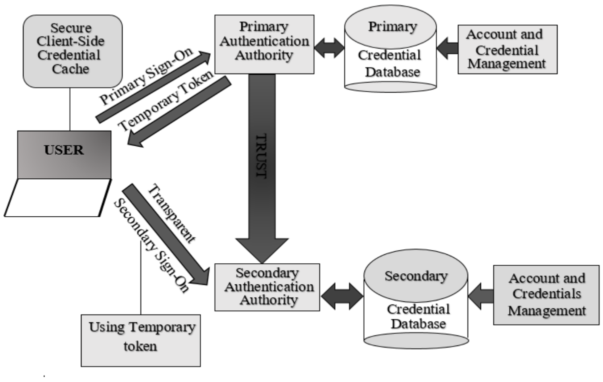
Secure Client-Side Credential Cache
Primary Authentication Authority
Primary Credential Database
Account and Credential Management
Primary Sign-On
Temporary Token
USER
TRUST
Transparent Secondary Sign-On
Secondary Authentication Authority
Secondary Credential Database
Account and Credentials Management
Using Temporary token

Fig 5: Token Based SSO

2.3 Trust Models of SSO

Different trust models need to be defined in order to evaluate various SSO solutions. These models vary depending on the scenario of business in which they are implemented. The model generally defines the various entities and their interaction and the overall system characteristics. Based on the services that the SSO environment support, three models have been defined. These are [4]:

1) **Authentication and Authorization Model (AAM):** AAM describes all the necessary frameworks that provide the basic two features that are authentication and authorization. The model being a traditional trust model represents the basic mechanism in which there is a service that checks all the users' credentials to decide whether an access should be granted or not to a user that is requesting an access. Here two major entities are involved: users that are requesting access to resources and services that share these resources. AAM model thus is based on a classic client-server architecture providing a generic protocol of authentication and authorization.

2) **Federated Model (FM):** It is one of the emergent trust model wherein several homogenous entities interact to provide the required services. Again here two major entities are identified: users that request access to resources and the services that share these resources. The major difference between AAM and FM model lies in the definition and composition of services. In the latter one, the services do not reside on the same domain. Hence are distributed on different domains that are built on the same level, thus allowing mutual trust and functionalities like cross-authentication are also being provided.

3) **Full Identity Management Model (FIMM):** This model is one the most challenging trust model and could merge the above two models. The model in addition provides mechanisms of identity and account management and privacy protection. Here three major entities are involved: users which request access to resources, services that share these resources and identity manager, which manages the user identities by providing the necessary functionalities. The model tries to fulfil the privacy needs that differentiates it with the previous models.

2.4 Different Types of SSO

There are three main variants of SSO: Web SSO, Legacy Web SSO, Federated SSO. We have given a brief description of each of them below [3]:

1) **Web Single Sign On:** Web Single Sign On is sometimes called as web access management. It enables a user to provide its credentials and only after the successful completion of authentication process, it establishes a relationship of trust that grants a user right to access all the resources for which he/she has been permitted.

2) **Legacy Web Single Sign On:** Legacy SSO is also termed as Enterprise SSO. After a successful authentication event, it manages multiple logins to specific applications. Web SSO and legacy SSO are almost identical in their structures. The difference lies in the fact that Web SSO only manages the web based service, while the Legacy SSO extends the SSO functionality to the traditional legacy applications and network resources, typically within an enterprise's internal network.

3) **Federated Single Sign On:** Federated SSO has a much broader concept than Web SSO. It uses Simple Object Access Protocol (SOAP) and Security Assertion Markup Language (SAML) to enable users to sign on once into a member of affiliated group of organizations and henceforth, access all the websites within that trusted federation. It extends the functionality of SSO from user's home domain to another foreign domain. This function of Federated SSO is its main advantage. Enterprises using federated SSO are allowed to maintain the control of their local services and the exposure of these resources to a larger class of users without the enterprise's direct administration.

3. SINGLE SIGN ON ENABLING TECHNOLOGIES AND PROTOCOLS

There are multiple protocols that can be used for SSO implementation like Kerberos, Security Assertion Markup Language (SAML) etc. Few of them are listed below [5]:

1) **Kerberos:** The Internet Engineering Task Force (IETF) has defined the Kerberos protocol as an open standard that is used on many platforms. The protocol makes use of Key Distribution Centre (KDC) as the server. It provides strong token based authentication using secret key cryptography for client/server applications. KDC authenticates the users to other servers for a particular session. The primary and secondary authentication domains share a trust relationship that is based on cryptographic methods and is used to validate the user token. The transportation of authentication tickets is done using Remote Procedure Calls(RPC). Kerberos is a good choice for organizations that wish to authenticate users using SSO to multiple applications across different technologies but the system applications should support Kerberos for this work.

2) **Lightweight Directory Access Protocol (LDAP):** The servers that centralize information about an organization such as employee names, employee address, telephone numbers and credentials are called directory servers and LDAP is used to query these servers. Active Directory a

Microsoft's version of LDAP enables true SSO using Kerberos but for Window's environment only. For application authentication, central LDAP is more practical than building authentication into each application.

3) **RADIUS Protocol:** The acronym stands for Remote Authentication Dial-In User Service. The protocol is used for authentication of remote users for example users that connect via Virtual Private Network (VPN). The protocol is a connectionless client/server protocol based on User Datagram Protocol (UDP). RADIUS server which is usually a daemon running on UNIX or windows machine when provided with the user credentials, can support various authentication mechanisms such as PAP, PPP or Unix login.

4) **Agent Scripts:** When security policies are revised or passwords are changed, scripts that run on a central authentication authority can be used to synchronize a user's password across systems. This can be done via Extensible Markup Language (XML) scripts and Structured Query Language(SQL) can be used for encryption to manipulate the data in databases.

5) **Cookies:** The pieces of software that are downloaded onto the client machine are called cookies. Cookies are token based SSO technology for HTTP environment that are used to authenticate sessions for certain time periods. The user will have to re-authenticate itself after the cookie expires.

6) **Digital Certificates and Public Key Infrastructure (PKI):** A system used for storing and maintaining encryption keys is referred to as a Public Key Infrastructure (PKI). This approach makes use of the public key cryptography for user authentication. The system relies on the role of Certification Authority (CA) for the issuance and management of the digital certificates and hence users' digital identities. The user first has to identify herself/himself to an authentication authority which issues a public key certificate to the authenticated user. Whenever the authenticated user wants to access a protected resource in subsequent authentication request, he/she creates a token and includes its digital certificate (public key) in it and signs it with her/his private key. On the reception of the request, the target server contacts the CA in order to verify the identity of the requesting user. There is a relationship of trust between the primary CA and the secondary CA as the latter's certificate is being issued by the former one. This enables any secondary CA to accept the certificate issued by the primary CA.

7) **Web Security Service:** It supports cross domain and cross platform communication among different business entities.

According to our study, the protocols used in web SSO are:

1) **Security Assertion Markup Language (SAML):** Security Assertion Markup Language (SAML) is a XML-based protocol developed by Organization for the Advancement of Structured Information Standards (OASIS). It is basically a platform independent, non-pro proprietary protocol that is used for communicating user identities between parties- who usually conduct business with each other. SAML is a key aspect of Federated SSO. It allows the communication between domains having different authentication mechanisms. The entities involved in SAML are the user, the Identity Provider (IdP) and the Service Provider (SP) as shown below. The IdP makes 'assertions' about the user's identity and attributes to the SPs. Service provider (SP) provides a specific service or hosts a target application.

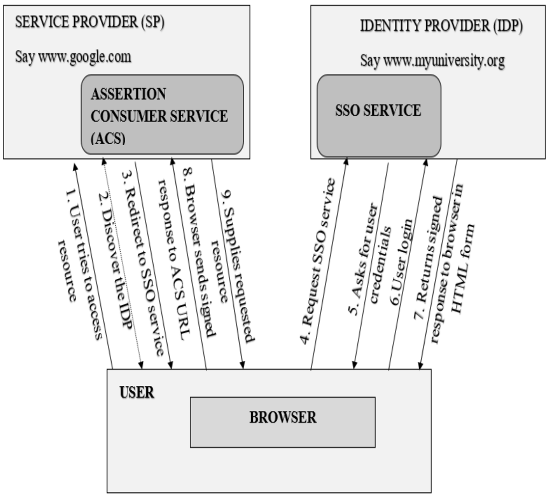

Fig 6: Protocol Flow of SAML

Figure 6 shows the flow of SAML. The three main entities are: user, Service provider and the Identity provider. The steps involved in the working of SAML are shown in Figure 6 where a user first requests Service provider like GOOGLE to access its application. The Service Provider can make a proper access control decision i.e., whether to perform a requested service for the particular user or not based on the assertion which is being provided by the Identity provider on request of the concerned Service Provider [6].

2) **OpenID:** OpenID is an open and promising user-centric Web SSO solution. A web SSO solution has separated the role of Identity provider (IDP) from that of Relying party (RP). An IdP collects user identity information and authenticates users, while for further authorization decisions, RP relies on the authenticated identity. In OpenID, the existence of trust relationship between IdP and RP is not required and users are free to choose or setup their own OpenID providers [7].

Figure 7 shows the protocol flow of OpenID. Its flow may vary with different implementations. Here, the user selects it's IdP or enters his/her own OpenID via a login form presented by a RP. The RP discovers IdP (OpenID provider) endpoint and redirects his/her OpenID to the IdP for authentication of the user. The user enters his/her username and password and hence authenticates itself to the IdP and then consents the IdP to the release of his/her profile. The IdP verifies the user's credentials and if valid, redirects the OpenID and profile signed by it to the RP. Ultimately, the user gets an access to the application he/she wished to access [7].

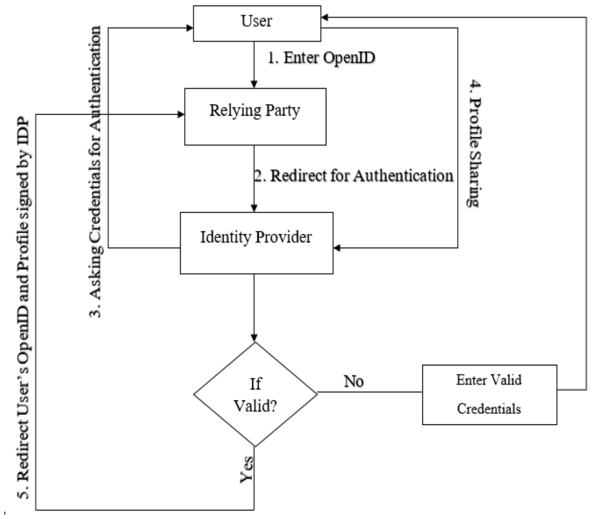
User
1. Enter OpenID
Relying Party
2. Redirect for Authentication
Identity Provider
3. Asking Credentials for Authentication
4. Profile Sharing
5. Redirect User's OpenID and Profile signed by IDP
If Valid?
No
Enter Valid Credentials
Yes

Fig 7: Protocol Flow of OpenID

4. BENEFITS AND DRAWBACKS OF SSO

The implementation of SSO has both positive as well as negative impact. The pros and cons of SSO are discussed below:

4.1 Benefits of Using SSO

Following are the benefits of using SSO [8]:

1) **Increased user productivity:** With the advent of SSO, users no longer need to memorize multiple IDs and passwords. Thus, SSO truly shouldered the burden of users by eradicating the hassle of multiple passwords. The users just need to go through single login step and enjoy access to multiple applications.

2) **Increased developer productivity:** The implementation of SSO provides a monotonous authentication framework to the developers. The developers need not to worry about authentication at all if the SSO mechanism is independent. When once a request for an application is accompanied by a username, the developers can assume that the authentication has already taken place.

3) **Simple administration:** The administration burden of user account management is also simplified when applications participate in SSO protocol. As SSO deals only with authentication, the level of simplification depends merely on the applications. Thus, some user specific requirements may still be required to be set up by the applications.

4) **B2B collaboration:** In today's world, most of the companies do not work alone but rather join hands and work as partners to bring out something fruitful. This collaboration of various businesses is of large scale and is functional only if the participating businesses are interoperable that is they should be able to interact with their disparate IT systems and should exchange data with ease. To make it happen, the enterprises make use of extranets. Here, enterprises allow their trading partners, service providers etc. to access not only their data but also some company owned applications. Thus, as a result various users need to perform multiple logins to various applications; some belonging to their own organization while some to their business partners. Hence, the authentication and authorization mechanism becomes quite hectic. The solution to this can be SSO. With SSO the enterprises can easily centralize their authentication management and allow users to login once and henceforth access all the shared applications. Thus, helps the collaborative partners to reduce production time, deliver better products on time [9].

5) **Security:** As the concept of SSO allows users to remember just one password, the users prefer to choose complex, hard to crack password instead of using multiple simple passwords. This improved the system security [3].

4.2 Drawbacks of Using SSO

The drawbacks of using SSO are as follows [8]:

1) **Scalability problem:** The SSO implementation can be difficult, time consuming and expensive to fit into existing applications.

2) **Logged in desktops:** Although the SSO implementation reduces security risks but the threats can be manifold. For instance, a legitimate user might sometimes just walk away from his system leaving his/her account logged in. A malicious user can easily gain access of it and hence all the authorized resources are compromised. Although, this problem can encounter with security generally, but the after effects with SSO are worse as without SSO only one resource gets comprised because separate logins are required for other applications.

3) **Single point of failure:** The arrangement is prone to denial of service attack as the authentication mechanism is centralized.

4.3 Challenges in the Growth of SSO

There are many challenges in the growth of SSO. Few of them are as follows [7]:

1) **Resistance to change:** Most of the users are comfortable with the existing system of multiple login system and don't want to switch from this traditional system to a new method of single step authentication. They mostly prefer to use the password manager feature of browsers.

2) **Security issues:** Most of the people are in an impression that by using single sign on mechanism, they are providing directly their usernames and passwords to the server and hence, their sensitive information is stored locally somewhere.

3) **Phishing issues:** The users also highlight phishing attacks as one of the main reasons that hinder the SSO adoption as they couldn't really find any distinguished difference between the real websites and the bogus ones.

4) **Trust issues:** This factor is the most crucial one. The users often hesitate to provide their personal and sensitive information to the websites using SSO. Often the websites using SSO feature ask the users to permit them to access their contact list, location and other sensitive information which clicks every users' mind before he/she agrees to the clause. Depending upon the popularity of the website and the trust relationship between the user and the website, user either adopts the SSO approach or decides to go for multiple sign-on sessions, thereby avoiding the situations where their information may get compromised somehow.

5) **Linking of accounts**: The users mostly are not familiar with the account linking process and when asked by a website to login by their existing social networking accounts, they get confused and frustrated and eventually drop the plan of SSO.

If the above factors are properly addressed, then the SSO technology may gain momentum in the lives of the socially active users. They may switch to the concept of “All the eggs in one basket”. Among all, the trust factor is the most significant one as the trust and privacy share inverse relationship i.e., as the trust increases, privacy decreases automatically. The websites following SSO mechanism should have proper design interfaces with least errors. If the users perceive a website to be honest, competent and convenient in delivering its services they would trust it and hence will easily accept the websites’ terms and conditions and will provide the correct and accurate information [7].

For all the information and resources kept in the system like personal detail information, users’ profiles, addresses, cost documents, certificates and policies related to the company, privacy has a role to play. These important documents must be stored securely so that the information will never get compromised. SSO identities carry the personal information of the user. Due to this reason, privacy is more important in the open SSO environments than the closed one. Several organizations are thus looking for SSO identities that do not carry personal details and support unlink ability feature for those identities which they are transporting inside the network. The proxy servers should be used to carry traffic between the users and the SP (Service Provider) in order to ensure that the user’s real network address is replaced by the proxy one [7].

4.4 Users Adaptability for SSO

Most of the users nowadays may like to switch over to SSO because of the following reasons:

1) **The Registration Challenge:** 86% of the people may leave a website when asked to create a new account due to way lengthy forms. 42% of the people find registration forms too lengthy, or ask too many questions.

2) **Password Fatigue:** 50% of the users dislike the idea of creating new password combination. Moreover, often the users already have to remember at least 5-6 passwords besides this new one. Almost 40% use the forgot password feature every month. Even some people even have a thought like solving the world’s peace problem is even easier than remembering so many passwords.

3) **Trust:** Trust is a problem. 88% of the users have admitted that they have lied on a registration form but 60% would give more information, if they knew how it would be used.

Though password manager feature shoulder the password fatigue burden but they can’t resolve the registration challenge. Then the choice users are left with is “Social Login”, introduced in 2008. Almost 77% prefer social logins. Social logins occur when users use their existing IDs from a social network or email provider. Facebook is the most popular one. 54% are using Facebook social login. For the first time since Janrain has been reporting, Facebook has exceeded 50% of the aggregate total of all social logins. Thus, the burden of registration is overcome by the concept of social logins. Most people like to return to a website if they get automatic recognition through social network [10].

5. COMBINATION OF (SSO) WITH MULTI FACTOR AUTHENTICATION (MFA)

The process of proving your identity and verifying that you are the same entity as you claim to be is referred as authentication. There are a variety of mechanisms available for authentication, these are: Biometrics, One Time Passwords, Digital Signatures etc. The authentication is called as Multi-Factor Authentication (MFA) if at least two of the three authentication mechanisms listed below are satisfied at a particular instance of time by a user [5]. These mechanisms are:

1) Something you know (a password or a PIN)

2) Something you have (say a smart card or a mobile phone)

3) Something you are (as represented by, say, a fingerprint)

An ATM based transaction is an example of Multi Factor Authentication as here two of the above three conditions need to be satisfied for a successful transaction. The ATM card holder must have his/her ATM for a transaction satisfying the type “Something you have” and besides this, he/she must have the pin code of the same which satisfies the another condition “Something you know”. Multi Factor Authentication if used in collaboration with SSO can help to reduce the security issues related to SSO and hence, making it more secure [5].

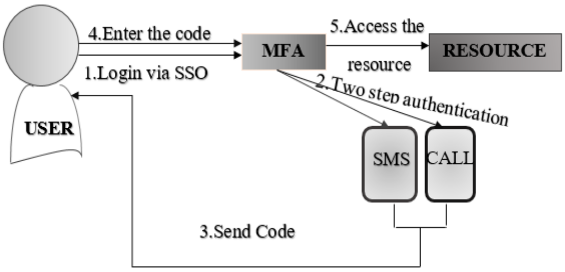

Fig 8: Combination of SSO with MFA

Figure 8 shows the combination of SSO with MFA where a user first signs in via Single Sign On approach (PKI encryption) and has to get through second authentication step using MFA like One Time Password (OTP) before getting access to the resource. OTP can be delivered to the user via SMS, phone call or using some mobile application. The time limit of the OTP is fixed after which the session expires making the code invalid.

5.1 Advantages of Using SSO with MFA

Following are the benefits of combining SSO with MFA [5]:

1) **Enhanced user productivity and user satisfaction:** The SSO approach increased user productivity and satisfaction as users need to spend less amount of time on logging into systems.

2) **Reduced IT costs:** SSO reduced the no. of password reset calls to the IT help desk up to 95% as now the users need not to memorize multiple passwords, it’s just one password and 90% of the users’ work is done. Furthermore, the installation, maintenance of separate authentication systems also is reduced as SSO provides central management of users. There is further reduction in IT costs and improvement in efficiency as the cost of

MFA systems like fingerprint scanners is decreasing but not at the cost of reliability.

3) **Improved Corporate data security:** The risk of a hacker getting "key to the castle" which was a major breakthrough in SSO is reduced by the coupled usage of SSO with MFA and also this implementation gives more assurance of a user's true identity- sometimes required for systems or transactions of higher risk to an organization. The attacks like phishing and malicious code attacks also can be prevented as though the password can get compromised somehow but the additional authentication factors like biometrics or tokens cannot be obtained easily.

4) **Enhanced business customer base:** The prime requirement of an account creation with a password often impede users from registering on retailer websites which can badly affect the business of owners. SSO via its technology of OpenID, allows users to register even without a password. Hence, encouraging more registrants and business.

5.2 Challenges in SSO/MFA Implementation

Some of the challenges associated with SSO/MFA implementation are [5]:

1) Although MFA minimizes security problems related to access control but still there are some security issues like the OTP tokens that can be compromised by man-in-the-middle phishing attack, smartcards can be hacked or stolen, the Trojan horses can be used by attackers to piggyback the user sessions after they have logged in. Though MFA can be cracked yet it does enhance the security of corporate networks rather that of Internet as the attackers keep on changing their tactics.

2) The success of SSO and MFA implementation relies on user acceptance. The users may resist to carry tokens with them or get fingerprinted. Tokens can easily get misplaced by users or stolen by attackers. Thus, user acceptance is a key challenge to SSO/MFA implementation.

3) Depending on the size of an organization and the type of SSO/MFA technology used, system costs can be extremely high. Some systems like client software may not support MFA devices and hence, new compatible devices (hardware and software) need to be purchased, configured and installed. Hence, a proper cost analysis can correctly determine the worth of SSO and MFA.

6. ANALYSIS

6.1 Is SSO a Security Practice?

The concept of Single Sign On has gained momentum in recent years with the increasing popularity of social networking. One login type applications provide access to tons of accounts across the panel, particularly in social media [6]. Most of the users prefer to use SSO approach because it is convenient for them as it allows them to access scores of individual accounts just by remembering a single password and henceforth saving them from the hassle of setting unique set of username and password to each of their accounts. Besides these benefits, SSO has some negative impact too but still the risks of SSO users are less than those of non-SSO users as the latter tend to keep almost same password to access their internal and external applications and mostly store their passwords in unsafe areas [11]. For example, if a person has three different accounts, he would prefer to set easy and almost same password to access each of its account say "his birthdate" or "his best friend's name" and so on.

SSO systems can centralize authentication on special servers as they are often based on complex systems management applications. Thus can make a positive contribution to an information security program of an enterprise. This can be done by using dedicated servers that hold SSO modules. These servers act as gate keepers, thus making sure that all the traffic trespasses through the SSO server first, which then passes along the credential it has stored for authenticating the particular application registered with the SSO system. To prevent malicious access, this centralization requires more planning, tuning and auditing than single authentication systems. The secure storage of authentication credentials and encryption keys in SSO systems make the hackers job even more challenging. Keeping track of users, pruning out inactive accounts of long-gone employees and monitoring suspicious activity are all part of SSO and can increase an organization's IT security [12].

Also by using the SSO in collaboration with MFA can help to reduce the risk of loss of "Single key to the castle" as for now even if the master key gets compromised somehow, still the hacker can't get access to the sensitive information of the user without crossing the second step of authentication [5].

Hence SSO approach altogether helps the users to easily access and manage different applications and services securely. Although there are some risks involved but if the user manages to keep the key secret, then SSO may be of great benefits.

6.2 Is SSO a Risk Factor?

SSO helps to avoid the loop of authentication but at the cost of some drawbacks**.** The users may be unable to authenticate themselves if the SSO provider goes down, thereby bringing the whole system's working to halt. This failure is called as single point of failure. Also, it is quite possible for a SSO server to get hacked or breached, which may lead to data loss. Furthermore, all of the crucial and confidential data of a user may get compromised in just a single shot, as all of the authentication credentials are in the same basket and the key to the basket may get revealed if the coupled usage of SSO and MFA is not implemented. Thus, it cannot be considered a total security tool. Sharing of user data with a third party is another underlying factor which enhances the risk factor of SSO usage. In order to cover a good portion of potential users, the right choice of Identity provider is vital. Hence, the disadvantages of relying on a third party is overwhelming and needs to be addressed to minimise the risk factor involved in SSO [12].

7. CONCLUSION

SSO is an access control method that allows a user to access multiple domains on a single step of authentication. Thus, relieves the user from the hassle of remembering numerous passwords for multiple applications. SSO is used for user convenience. However, if the main key of the authentication is breached, then the user's crucial details may get compromised. As discussed, this threat of losing master key can be reduced by using SSO with MFA. The combination of SSO and MFA may allow a user to not worry about the negative consequences of losing his/her master key. Due to second step authentication, an intruder cannot access a user's confidential data by acquiring just a key. However, MFA also has a bottleneck. It can get compromised by phishing or man-in-the-middle attacks. For example, a fraudster somehow may steal a legitimate user's OTP which is possible by installation of malicious software on the target's device and hence, bypassing the traffic meant for it. However, these threats can be addressed by several ways like disabling key logging and screen capturing of OTP, securing authentication and communication against malware attacks etc. Thus, the benefits of using SSO are worth mentioning and if its disadvantages are properly addressed and tackled, it will be user convenient and will help them to access applications with ease and security. As a part of future work, we would like to focus on implementing security mechanisms in combination of SSO and MFA.

8. REFERENCES

[1] Li, B., Ge, S., Wo, T. Y. and Ma, D.F. 2004. Research and Implementation of Single Sign-On Mechanism for ASP Pattern. In Proceedings of the Third International Conference on Grid and Cooperative Computing.

[2] [Online]http://blogs.vmware.com/vfabric/files/2013/03/authentication_chart.png

[3] Patil, A., Prof. Pandit, R., and Prof. Patel, S. 2013. Analysis of Single Sign on for Multiple Web Applications. J. Advanced Research in Electrical, Electronics and Instrumentation Engineering, (Aug. 2013), 4104-4107.

[4] Ardagna, C. A., Damiani, E., Vimercati, S. C., Frati, F.and Samarati, P. 2006. CAS++: an Open Source Single Sign-On Solution for Secure e-Services. In Proceedings of the 21st International Information Security Conference on Security and Privacy in Dynamic Environments.

[5] Alphonso, M. D., and Lane, M. 2010. The Adoption of Single Sign-On and Multifactor Authentication in Organizations – A Critical Evaluation Using TOE Framework. J. Issues in Informing Science and Information Technology, (May 2010), 172-184.

[6] [Online]https://wiki.eclipse.org/images/e/ec/Saml2idp-1.png

[7] Sun, S. T., Pospisil, E., Muslukhov, I., Dindar, N., Hawkey, K., and Beznosov, K. What Makes Users Refuse Web Single Sign-On? An Empirical Investigation of OpenID. J. Symposium on Usable Privacy and Security (SOUPS), (Jul. 2011), 1-3.

[8] "The Advantages and Disadvantages of Single-Sign-On (SSO) Technology", *Secure Connexion*, 2012. [Online] https://secureconnexion.wordpress.com/2012/08/24/the-advantages-and-disadvantages-of-single-sign-on-sso-technology-mini-whitepaper/.

[9] Villanueva, J. 2014 "5 Big Business Benefits of Using SSO (Single Sign-On)" Managed File Transfer and Network Solutions.

[10] [Online]

http://janrain.com/wp-content/uploads/2012/10/how-to-solve-the-online-registration-challenge.png

[11] "Does single sign-on (SSO) improve security?" *SearchSecurity*, 2016. [Online] http://searchsecurity.techtarget.com/answer/Does-single-sign-on-SSO-improve-security.

[12] Davis, M. 2013 "The Pros And Cons Of Single Sign-On for Web Services", *Future Hosting*, [Online] https://www.futurehosting.com/blog/the-pros-and-cons-of-single-sign-on-for-web-services/.