International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 04 | April - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Cryptography Based Single Sign On

Bhavin Merai [1], Chinmay Vernekar[2], Himanshu Bist[3], Siddhant Shetty[4]

[1]BE EXTC & St John College Of Engineering And Management
[2] BE EXTC & St John College Of Engineering And Management
[3] BE EXTC & St John College Of Engineering And Management
[4] BE EXTC & St John College Of Engineering And Management

Abstract - Data breaches and stolen credentials is an issue for both service providing companies and the users using the services. This issue can be due to improper handling of sensitive data from user side or company's side. Single Sign On is a web service where this sensitive data such as user credentials is handled by valid central authenticating bodies. In this paper such a single sign on application is exhibited and one of the multi factor authentication methods is added to this application to extend sign in towards mobile devices.

Key Words **- single sign on, hash, authentication, protocol**

1. INTRODUCTION

According to the 2019 midyear estimates of the world internet usage and population statistics, there are more than 4.5 billion internet users in the world, and this number is persistently increasing. From 2000 to 2019 the number of internet users increased by 1157%, this significant increase in the number of users can also be attributed to the remarkable developments in network technologies [11]. With the expanding number of internet users, businesses want to provide them with a rich and varied digital experience. Websites, online apps, and web services are used to create this web experience. Users are required to sign up for websites in order to engage with them and use their services. The most valuable aspect of every website is not the application's service or the algorithm for which it is employed, nor is it the website's user experience; it is the user's data and credential sets, which are assigned the biggest priority by both users and companies. Unfortunately, when it comes to data breaches prevention is never 100% possible. There are ways to reduce the chances of users having their credential sets stolen in a breach. There are ways to reduce the chances of users having their credential set stolen in a breach. One of such approaches is termed as Single Sign On. Single Sign On is an authentication scheme that allows a user to log in with a single ID and password to any of several yet independent software systems. True single sign-on allows the user to login once and access several services without re- entering authentication factors. Single Sign On (SSO) is a mechanism that enables a legal user with a single credential set to be authenticated by multiple service providers in a distributed computer network. With the widespread use of distributed pc networks, it has become common to permit users to access numerous network services offered by distributed service providers. Consequently, user authentication (also called user identification) plays an important role to verify if a user is legal in distributed computer networks and can so be granted access to the services requested. To avoid phony servers, users sometimes ought to attest service providers [6]. The user being a human has to memorize different credentials for multiple services used. They have to remember the credentials correctly to avoid seamless usage of services for that particular application on the web. By analyzing the practical situation, this method is very difficult for a user to maintain the records for these sets of credentials for various service providers. It increases the processing workload on the users and these service providers. To overcome this problem, the single sign on scheme is introduced.

2. EXISTING PROTOCOLS

a. Kerberos

Kerberos was one of the first single sign-on solutions proposed in the literature and implemented as a network service. It is formally described as a network authentication system, initially designed for providing single sign-on to network services. A Kerberos infrastructure is composed of four entities.
Client.

1.Authentication Server.
2.A Ticket Granting Server.
3.An Application Server.

b. SAML

SAML (Security Assertion Markup Language) is an open standard for sharing authentication and authorization data between security domains that is based on XML. An online service provider uses SAML to communicate with an online identity provider, which authenticates users attempting to access sensitive content. SAML does not explain how to authenticate a user; rather, it specifies how to share

authentication and authorization data once the user has been verified.

c. Open ID

OpenID, which also provides a framework for enabling flexible centralized user authentication for web-based services, is one of the most successful commercial single sign-on solutions. The user can select from a number of identity providers in OpenID, including any website or web-based application where he already has an account (e.g. Google). OpenID consists of three main entities:

1. The OpenID Identifier.
2. The OpenID Relying Party (RP)
3. The OpenID Provider (OP)

d. OAuth

OAuth is an open-standard authorization mechanism or framework that enables "safe designated access" in applications.

The flow of OAuth 2.0 protocol includes the following steps:

1. The client requests the resource owner directly or indirectly via the authorization server for authorization.
2. The client gets an authorization grant.
3. The client makes requests for an access token. This is done by authenticating with the authorization server and also presenting the authorization grant.
4. The authorization server authenticates the client, validates the authorization grant, and on successful validation, issues an access token.
5. The client authenticates himself via the access token. The client then requests for the protected resource from the resource server.
6. The resource server validates the access token, and upon successful validation, responds to the request.

e. Quick Response Code

Quick Response (QR) is a type of barcode easily readable with digital devices like smart phones. They store information as a series of pixels in a square grid that can be read in two directions - top to bottom and right to left - unlike standard barcodes that can only be read top to bottom.

The Authenticate with QR code (authentication token) API is similar to the central Authentication API in that it is used for user and device authentication.

The authentication tokens endpoint lets you verify a user's identity by scanning a QR code or, if you're using a mobile device, by clicking on a link.

The following are the general steps in the flow:

1. Create an authentication token, and the authentication tokeXID will be generated as well.
2. Incorporate the token into a QR code image. If the user is using a mobile browser, you can build a link.
3. The authentication token ID is used to poll the authentication token. The returned token status will be updated once the user has successfully used the token, either by scanning the QR code or clicking on the link.

3. APPROACH

The existing protocols outlined earlier in this paper are employed by various single sign on providers in the market. These protocols have been developed after intensive research over a long period of time focusing on security with the growing usage of web 2.0. The approach followed in the development of the single sign on application is established on one of the mentioned protocols i.e. OPEN ID and OAUTH. One of the important features added to the application developed also uses another of the mentioned protocol i.e. Quick Response (QR) Code.

4. IMPLEMENTATION

A. Single Sign-On Mechanism

Single sign-on (SSO) is an authentication scheme that allows a user to log in with a single ID to any of several related, yet independent, software systems. Single sign-on also allows the user to log in once and access the services without re-entering authentication factors repeatedly.

The various phases in which the method is carried out:

1) Login Phase

2) Registration Phase

3) Authentication Phase

4) Shared Session Phase.

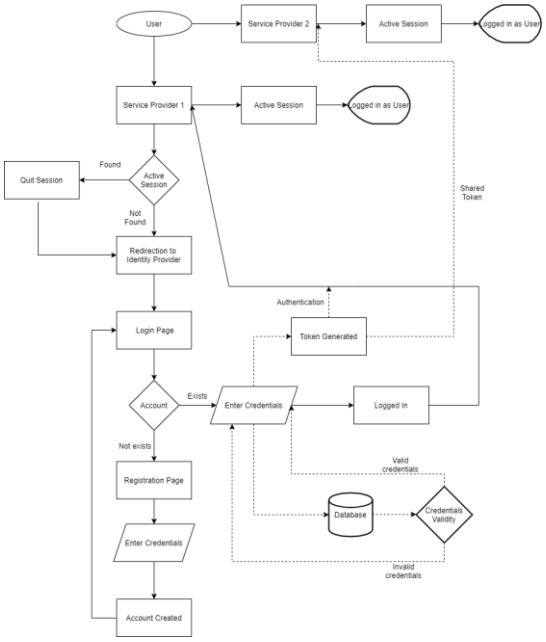

Fig. 1. Single Sign-On Mechanism Fl

1) Login Phase- In the login phase of the single sign on mechanism, user/client goes to any application and tends to login into that application to use its services. It is considered that this user is a new user who has not used the application or connected with the service provider. Since the user is termed as a new user, the user has to create an account with an identity provider which is a sovereign body connected with this service provider and many others to provide a user the consent to use all the services after it proves himself as a valid user. In the application when the user kicks off the login process on the application by browsing towards the login page of the website of this application, the user gets redirected towards the identity provider page over the web.

2) Registration Phase- In this phase, after getting redirected towards the identity provider over the web, the user is presented with a login page with a facility to register as a new user if the user has never connected with the identity provider. Considering the user has never connected with this identity provider, the user selects to register, where this user has to enter various information pieces which prove the validity of the user using this application in a real-world scenario. After getting registered the user has obtained a set of credentials including a password which he uses over the login page which he had come across earlier, since after registration the user again gets redirected to the login page of the identity provider. In the backend of this whole process, the user credential sets are stored over a database which is linked with the identity provider and the Idp requests this database to check whether the information entered while login are already present in the database. If not, the user receives an error. If the credential sets get matched the user gets a successful login and enters into a session. The credential set includes a password which is used by the user frequently while getting logged in. This password proves the further validity of a user so it is a piece of information with great importance. So while getting stored into the database this password is encrypted by going through a hash.

3) Authentication Phase- The authentication phase of the application for the user takes place over the backend. After the user enters the session mode through the Idp in the backend a token gets generated. This token is a JWT. The token is in an encrypted Json format. It contains a header, a payload and a signature. The header specifies the algorithm used, payload contains pertinent user information and the signature is a hash. The token has got the credential sets used for logging in and the algorithm used to encrypt it in a Json format. It also includes a timestamp which records the date and time of the login process. This token is always a time limited token meaning that it is valid for only certain duration. This duration is set by the developers while developing the Idp application. It can only be changed by the developers. The user information, timestamp and other necessary information is always inside the payload.
Due to the credential sets and timestamp being encrypted along with time duration for the token validity, these tokens cannot be shared over the web to be used by other users and the same session cannot be started after the expiry. The signature which gets generated after hashing of the header and payload and inserted into the token is what ensures that contents of the token were not tampered during token transmission.
The application which tries to validate the JWT to grant access to the user, uses a pre-shared key (or public key based on algorithm used) to hash the header and payload and match this hash with the signature that is already inserted in the token. If both hashes match, the user gets authenticated. This token authenticates the user over various Sp applications. Also after entering this phase the user has already been directed to the application which was started earlier at the start of the process. On that application website, a session of the user can be seen which would be generated by the Idp.

4) Shared Session Phase - In this phase, the user has already been surfing through the application he required at the beginning. The user now wants to surf through another one of the applications which has already been present in the domain of the Idp. This means that the other application is connected to the same Idp that the user has already logged in. In this case, the user is not again redirected to the Idp. Instead the user can see that the same session is running already considering that the JWT token is not expired. If the token gets expired, the user might be redirected. The same session running over the other applications in the Idp domain is achieved through a shared session technique. The token which got generated after entering the session phase of the Idp, contains an encryption algorithm which is used to encrypt information in Json format. This forms a signature in the Json token and when the user directs to another application a hash comparison of the signature validates the token. Until the token timeout each web service connected with the same Idp requires only a pre-shared key (or a public key based on encryption algorithm used) to validate the token.

B. Keycloak Server

The application developed over the single sign on mechanism implemented earlier is deployed over a custom developed authorization server. This authorization server is developed using Java Spring boot tools which a framework is helping to develop web applications and other micro services over the web. This method of development of a

custom authorization server for each single sign on schema helds out as an outdated process in the current web 2.0 world. This does not allow the current protocols such as OpenID and OAuth 2.0 to be configured with the server. So to overcome this problem thereis a need to deploy this

application over a server which is bothreadily available and open source. That's where the Keycloak authorization server comes into picture. KeyCloak is an open-source Identity and Access Management arrangement focused on present day applications and administrations. It makes it simple to protect applications and administrations with almost no code. It offers an expansive arrangement of highlights, as SSO, verification and approval, social login, multifaceted confirmation and concentrated client the board. Clients validate the KeyCloak server and don't have to verify various applications, utilizing SSO innovation, depend on standard conventions and offers help for OpenID Connect, OAuth 2.0 and SAML. Among the many features of KeyCloak include:

1. Admin Console to arrange the KeyCloak server and make domains, jobs, clients and customers.

2. Single Sign-On (SSO) utilizing the OpenID Connect (OIDC) confirmation convention on OAuth 2.0.

3. Client Adapters to incorporate Spring Boot, Spring Security and Angular with KeyCloak.

4. OpenID Connect support.

5. OAuth 2.0 help.

6. Social login: enable login with Google, GitHub, Facebook, Twitter and other informal organizations.
Through the admin console administrators can centrally manage all aspects of the Keycloak server. They can enable and disable various features. They can configure identity brokering and user federation. They can create and manage applications and services, and define fine-grained authorization policies. Users authenticate with Keycloak rather than individual applications. This means that your applications don't have to deal with login forms, authenticating users, and storing users. Once logged-in to Keycloak, users don't have to login again to access a different application.
When a confidential OIDC client needs to send a backchannel request (for example, to exchange code for the token, or to refresh the token) it needs to authenticate against the Keycloak server. By default, there are three ways to authenticate the client: client ID and client secret, client authentication with signed JWT, or client authentication with signed JWT using client secret.
In order for an application or service to utilize Keycloak it has to register a client in Keycloak. An admin can do this through the admin console (or admin REST endpoints), but clients can also register themselves through the Keycloak client registration service.
The client registration service provides built-in support for keycloak client representation, openid connect client metadata. The client registration service endpoint is /auth/realms/<realm>/client-registrations/<provider>.
They can also manage users including permissions and sessions.

C. Multi-Factor Authentication

Use a proportional serif typeface such as Times Roman or Times New Roman and embed all fonts. Table I provides samples of the appropriate type sizes and styles to use.

The displaying of two or more of the three authentication factors in a web framework i.e. A knowledge factor, a possession factor, and an inherence factor, are known as multi-factor authentication.
In multifactor authentication schema to get authenticated, the user is required to provide his credential sets along with other kinds of verification factors. These additional verification factors are provided by the central validating body. One of such multi factor authentication methods is Quick Response codes. These are a two-dimensional matrix styled barcode which encrypts and stores information. QR code generation for encryption could be applied to multiple services. The QR code can be considered as a visual representation of a onetime password. The QR code can be scanned by the users through their handset devices. The QR code can store data in top to bottom and right to left manner as compared to traditional barcodes which stores data in one direction i.e. right to left. The QR codes themselves can't be hacked; the security issues connected with QR codes are caused by the codes' destination, not the codes themselves.
The patterns in QR codes are binary codes that may be decoded to expose the data contained within the code. The three huge squares outside the QR code can be used by a reader to identify a normal QR code. When the three squares of QR get recognized, the reader understands that everything inside the QR code is data. The QR code is then analyzed by breaking it down into a grid by the reader. It examines each grid square and assigns a value to each one based on whether it is black or white. The grid squares are then grouped together to form larger patterns. A standard QR code is identifiable based on three components:

A. Finder Pattern- QR codes normally have three black squares in the bottom left, top left, and top right corners as a finder pattern. These squares indicate to a QR reader that it is seeing a QR code and where the code's outer boundaries are located.

B. Alignment Pattern- Another smaller square can be found towards the bottom right corner of the alignment pattern. It ensures that even if the QR code is twisted or at an angle, it can be read.

C. Timing Pattern- Between the three squares in the finder pattern, there is an L-shaped line called the timing pattern. The timing pattern aids the reader in identifying particular squares within the entire code and enables the reader to interpret a broken QR code.

Although QR codes can be used for a variety of applications, there are four widely acknowledged variations, the "input mode" governs how data can be saved and is determined by the version utilized. It can be either numeric, alphanumeric, binary, or kanji. The version information section in the QR code communicates the mode type.

A. Numeric mode - This is the mode for decimal digits 0 through 9. With up to 7,089 characters accessible in numeric form, it is the most efficient storage mode.

B. Alphanumeric mode - This mode accepts decimal digits 0 through 9, as well as uppercase letters A through Z, as well as the symbols $, percent, *, +, –, /, and:, as well as a space. It has a storage capacity of up to 4,296 characters.

1. Byte mode- Characters from the ISO–8859–1 character set are used here. It has a capacity of 2,953 characters.

2. Kanji mode - This is for the Shift JIS character set's double–byte characters, which are used to encode Japanese characters.
This is the original mode, which Denso Wave invented. However, with just 1,817 characters available for storage, it has now become the least effective. Extended Channel Interpretation (ECI) mode is a second kanji mode that allows you to define the UTF–8 kanji character set. Some newer QR code readers, on the other hand, will be unable to read this character set.
The application was created using the single sign on mechanism, which included the creation and sharing of JSon web tokens for user authentication across multiple web services, as well as deploying the entire application on the open source keycloak server.
The feature of QR code generation was added to the application that would allow users to sign in using their mobile or handheld devices. The web token of the session along with the user credentials and the timestamp will be embedded in the QR code. This could be scanned from the dedicated mobile app to avail the web session over a mobile device.

5. Result

The implementation exhibits a proposed system methodology based on which an application was developed. This web application has two client/service providers and these are connected to a central identity provider. All of these are deployed over the keycloak server. This running application demonstrates the whole single sign on mechanism.

6. Limitations And Discussions

As stated earlier the development of a custom authorization server possessed its own sets of limitations and the path of deploying the whole application over an open-source server i.e. The Keycloak server was taken while developing the application. But this server comes up with its own pre-defined regulations. There is no scope of adding custom key sets to both kinds of server. The open- source server has its own certificates and encryption algorithm and key sets which have to be used while deploying the application. Also up to this day there is no technology developed in the single sign on applications segment which provides a way to open and/or continue the same session over different web browsers. A patent has been filed by Google which ideated over this cross- browser session interface but there is no application readily available which can provide such an interface.

7. Conclusion

In this paper, a method on development of an application based on the Single Sign On (SSO) scheme for protection of users privacy and authorize users to use the different web services is manifested. The proposed framework enables smooth and transparent single sign-on without compromising overall network security or necessitating any online interactions between service providers and identity providers. The paper also showcases the limitations still faced in this domain to be open for more research to be done and applications that can be further developed on the single sign on schema.

8. References

[1] Implementation of Single Sign-On Mechanism for Distributed Computing – June 2014.

[2] Single Sign-On Mechanism Using Attribute Based Encryption in Distributed Computer Networks-2015.

[3] Security Enhancement of a Single Sign-On Mechanism based on ECC based Verifiable Encryption of Credential –May 2014.

[4] Single Sign-On Mechanism for Multiple Social Media Sites, Vol-2 Issue- 62016, IJARIIE-ISSN(O)-2395-4396.

[5] Guilin Wang and Jiangshan Yu, Security Analysis of a Single Sign-On Mechanism for Distributed Computer Networks, Center for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia.

[6] Single Sign-On Mechanism with Enhanced Security Measurements- March 2014.

[7] Analysis of Single Sign on for Multiple Web Applications.

[8] A Framework for Secure Single Sign-On Department of Electrical Engineering University of Brasilia, Brazil 2011/246.

[9] A Single Sign on Based Secure Remote User Authentication Scheme for Multi-Server Environments.

[10] Implementation of Rivest Cipher 4 algorithm in Security Assertion Mark- up Language protocols on Single Sign-On services 2021.

[11] Security and Performance of Single Sign-On Based on One-Time Pad Algorithm December 11-14 2014.

[12] A Review on Identity and Access Management Server (KeyCloak), Artech Journal of Effective Research in Engineering and Technology (AJERET), Volume, 1, Issue 4, 2020, Pages: 104-109, ISSN: 252 -61.