INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078

(Volume 9, Issue 2 - V9I2-1229)
Available online at: https://www.ijariit.com

MO-Auth: A Novel Approach for Authentication in Modern Applications

Syed Taimoor Ali
rizvisyedtaimoor@gmail.com
Central South University, Changsha, China

ABSTRACT

Technology has brought massive evolution in the application development process. The trend of online business has enhanced interest in migrating organizations towards modern technologies. Service providers, vendors, and clients are worried about security when valuable credentials of the organizations are placed online over the web. Therefore, security protocols and cryptographic techniques are used to prevent vulnerability. Several authentication techniques have been used to secure information, this article describes some of them. The concepts from three famous techniques were adopted to practice and propose a novel authentication method MO-Auth. Experimental work was conducted by implementing two web-based applications on which MO-Auth was tested. To validate results ISO/IEC 9126 quality characteristics were evaluated for each authentication method including MO-Auth. By considering ISO/IEC 9126 quality frameworks, we have found MO-Auth as easy to use, reliable, efficient, less cost, easy to maintain and much secure authentication method. Moreover, we have also conducted a survey-based on available features in modern applications, which indicates that usage of traditional authentication techniques is not decreased but still being used with the collaboration of modern technologies, while methods like OAuth and Biometrics are newer but their usage will be increased in future.

Key Words: *– Authentication; Web Security; Web Technology; Information Security*

1. INTRODUCTION

Since few years' technologies have grown and evolved. In the scenario of modern application development, service providers are focused to enhance business acumen and provide a variety of services to clients for achieving their satisfaction. Applications have heterogeneous clients over the internet and modern web applications have enabled clients to make interaction with servers and perform CRUD (Create, Reed, Update and Delete) operations from GUI using HTTPS protocols. In that scenario service providers

must ensure the security of information over the web. Security and privacy are key modules of software applications, these modules are built to secure users and their data by adopting highly security protocols. When users are using applications, it is required to trust the servers, so that they can keep the necessary data secure from unnecessary robbery, the confidence on systems can often be wrong because there are many tricks in the present era with the help of secrets data can be fetched by an unauthorized person. Application developers are enforced to apply high encryption techniques on the systems to identify users and provide them appropriate functionalities. In some applications, users are categorized and served with specified services. For that scenario, there must be a mechanism that performs isolation of user and system according to the role. Elasticity among applications is increased in the cloud environment, it enabled clients to work in a collaborative environment and move between distinct applications frequently. In that scenario, the security modules are aimed at a priority. Authentication is a process to identify users and provide them access to the services in the system. Modern applications have multiple authentication techniques, this study is conducted to enhance knowledge about authentication securities of modern applications. In the experiment a CMS (Content Management System)-based web prototype was implementing; concepts were adopted from three widely used authentication methods to propose a novel methodology for authentication.

2. RESEARCH CHALLENGE

Modern applications are moving towards new trends to make business more efficient and cost-effective. Illegal interruption/interception on an application can affect the system, modification or operation performed by the unauthorized user can ruin system functionalities. To resist these cyberattacks, vendors must provide a highly encrypted authentication system in applications to secure data and transactions. In Applications, mechanisms for data maintenance and single sign-in are more cost consuming, to solve this problem several authentication methods were proposed and still being used. Some of these techniques are described below in the methodology. Furthermore, this study also proposes a novel authentication approach MO-Auth.

3. RELATED WORKS

Authentication Methods should be designed well, what if the failure occurs in the authentication of a valid user, online resources on the internet will be left at risk which may be misused by an unauthorized user. Therefore, the author discussed semantics methods and mechanisms for authentications and explained the aims, operations, and limitations of the various authentication systems [1].

In current scenario organizations are moving towards technologies, organizations have their internet applications therefore, security is the main concern. Many authentication standards are being used and O-Auth is one of them. The author conducted a survey and discussed the mechanism of an open authentication method, in results he claimed that the process of authentication is reduced by adopting OAuth trough which usage of internet applications became easier [2].

Organizations or industries are getting much interest to migrate to modern technologies. For that, it is necessary to think about usability, maintainability, and security of important credentials. The author considers the security framework as a priority and did a survey on current authentication and authorization systems in applications [3].

The trend of Single Sign-On (SSO) or Open ID system like O-Auth is increased in cloud computing, and service providers are focused on security in systems. Server-side encryption with an open cloud is dangerous, therefore encryption at the client-side can be considered as an advantage for the cloud. The author proposed a client-side validation pattern for distrusted systems where clients are verified by Session, Java Mail API and Public key foundation [4].

Biometrics systems are replacing traditional authentication methods, the author conducted a review study to find an improvement in reorganization accuracy and security. The author analyzed and discussed existing research works, the role of biometrics in modern application security challenges in biometrics like accuracy in scanning, pros, and cons of biometrics and more, he also gave some suggestions for future work [5].

4. METHODOLOGY

Several security methods are discovered by researchers, in this study we have discussed some of these by considering different frameworks. Moreover, we did a survey on famous leading web sites and their authentication methods. In the end, we have adopted

three authentication methods to propose novel method MO-Auth. We have implemented MO-Auth in our designed web-based prototype to meet with the result.

4.1 Encryption

Most of the security controls in web applications are dependent on cryptography, which often used to secure sensitive data and enhance the security of systems. Encryption is a technique that transforms all user transactions into the unreadable format and prevents non-authorized users to make interference. Encryption is used when clients are providing personal information on applications like; credit card numbers, home addresses or social security keys. The process of encryption is based on keys, and encrypted data cannot be decrypted unless the key is provided.

Some application does not encrypt the data while the client is communicating with the server. But securing data is essential in applications to ensure clients that transmitted information is secured and encrypted end-to-end. Therefore, it is so important to use encrypted protocols and establish new encrypted sessions or use IPSec tunnel while data is transferred. Request from a client is transmitted to the servers, usually, both web applications servers and database servers are placed on the same trusted network but the use of SSL is essential to encrypt client-server communications. Encryption is about ensuring confidentiality which lets only an authorized recipient read and write the data. The Secure Shell (SSH) and Socket Layer (SSL) protocols are usually used in encryption processes. SSL is now known as Transport Layer Security (TLS) we can refer them as SSL/TSL, SSL/TSL operates the secure part of HTTPS sites [6]. Data in SSH sessions and SSL/TSL is encrypted between client and server unless the connection is stabled.

Fig. 1. Client-Server Communication Over HTTPS [6]

Fig. 1. Web applications use public-key cryptography to create a shared session key. It then communicates through symmetric key cryptography using this shared session key. Public key cryptography is a famous protocol for online applications because the user has no need to disclose private keys to anyone, that secrecy decreases the chances of cybercriminals to tackle transmitted data.

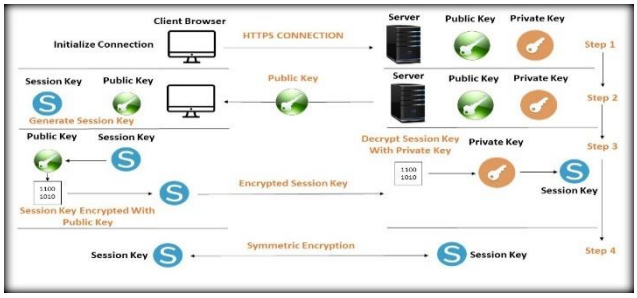

Fig. 2. HTTPS Encryption [7]

Fig. 2. Illustrates the scenario of encryption over HTTPS protocol, initially, the client establishes the connection from the browser by sending a request to the server. In step 2 server sends the public key as a response to the client and keeps private key secret, in step 3 browser generates a session key and encrypts it with a public key that is provided by the server and sends it to the server. The server decrypts it with a private key and terminates public-key encryption by replacing it with symmetric encryption. At step 4 server is using symmetric encryption with sessions, and the life of sessions will remain until the client leaves an application.

4.2 Authentication

Network entities do not have physical access to the clients, so identifying valid clients over the web is essential to secure application. Authentication is the process to identify valid users over an application and enable them to have appropriate access for available services.

Authentication is essential for the application to keep the network secure by preventing illegal activities over the system. Several authentication methods have been founded by researchers to provide right access control to the valid recipient, these methods may relate to three major factors [1] which are described as:

- ***Something You Know***

It is authentication in which secrete keys are familiar to the human mind; like Personal Identification Number (PIN) or Password-based authentication.

- ***Something You Have***

It refers to information which client can physically carry with himself. A piece of information used once, which may expire after one use or expire within define time. Token-based password, ID Cards, Smart Cards can be considered as an example of "something you have".

- ***Something You Are***

The user itself is considered as a security key. Characteristics that only exist in the user can be used as a password. Biometrics, retina detection, voice, and face detection are examples for "Something you are".

TABLE I
PROS AND CONS OF AUTHENTICATION FACTORS [8]

Type	Advantages	Disadvantages
Something You Know (Knowledge-based)	Low cost & Convenient	Hard to find if lost
Something You Have (Token-based)	More Secure, hard to copy	Involves additional costs, such as the cost of the physical token and any replacement fees.
Something You Are (Characteristic-based)	Hard to steal, can't be forgotten, always available	Cannot be replaced once compromised, required expensive hardware

4.3 Password-based Authentication

Basically, passwords are composed strings of alphabets, numbers or special characters, which are used as a security key to avoid vulnerability and get the right access on application. Password-based authentication is a simple way to authenticate valid users over the web. Username and password are stored once on the server, the system administrator is tracking of both username and password for each user on servers [8].

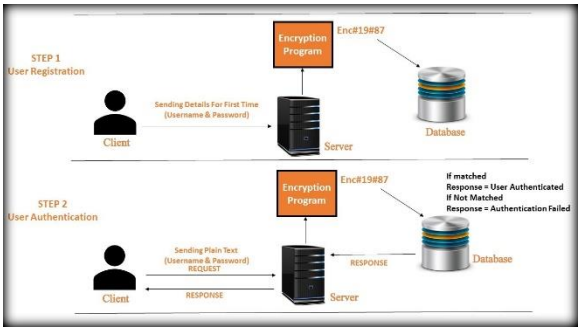

Fig. 3. Password-based Authentication [8]

When the user makes registration to the system for the first time, provided data is sent to the server and the server forwards it to the encryption program for converting data in the unreadable format and finally stores it into the database. The user has already trusted the server by making registration on the system. Now for the authentication process, a user is placing username and password in given fields at GUI. Provided username and password are sent to the server by the network, and the server determines either provided information is valid or not. If the provided information is matched, then the server will grant the user access to application else user will be redirected to the same web page to put valid information [8]. Authentication using a password is easy to use, easy to change, easy to implement and low-priced. It does not require additional hardware equipment, just a single program can handle mechanism for the number of users.

The security of the user is totally based on password strength. A weak password can be predicted by un authenticate users, that's why the user is always recommended to provide a strong password. Through Phishing and password prediction methods unauthenticated can access the system.

4.4 Token-based Authentication

Token-based authentication is the most commonly used one-time password method. It often used to provide additional security with a password. Basically, it is a small device placed at the server and preprogrammed with random seed or unique numbers to generate random values for the password every time it is used.

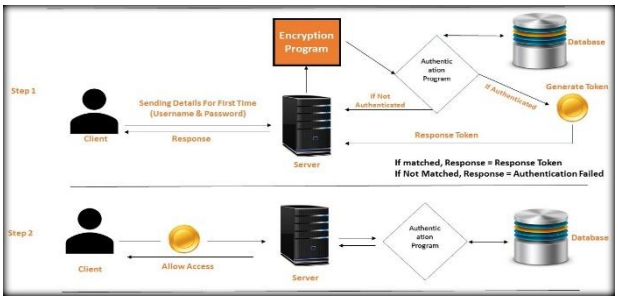

Fig. 4. Token-based Authentication [9]

Fig 4. Illustrates the mechanism for token-based authentication. It is used to enhance the security of an application [9]. When the user is trying to get access to the application by providing username and password, the request is sent to the server, and the server executes an authentication process by compares the provided username and password with information stored in the database. If given information is matched, the unique token is generated, which is an additional key to access an application. The generated token is forwarded to the client as a response by an email or contact number. Now the client sends that token to the server and the server determines token whether it is authorized or not, if yes server will allow system access to the client.

If the generated token is used once, the system will have to repeat the whole process again for the new login. Once used token cannot be used again to authenticate the user. Some programmer uses a time-based token for authentication. In which generated token is created for a specified time duration. If a token is used within the defined time then the user will be authenticated, else if not used within time, a token will be expired and will not let the user be logged in.

Token-based authentication is more secure to use than other authentication methods because it works as an additional layer of security. It uses third party certification by verifying token via email or SMS, which enhanced the business of an organization by securing user credentials more effectively.

The drawback of token-based authentication is that it increases entities in database tables to maintain digital certificates like seeds, so the cost for maintaining token seeds also increases. Expenses for using third party gateway are also increased.

4.5 Authentication Using Biometrics

Biometric is a new trend to authenticate valid recipients over the system. An authentication process is done by measuring the physical characteristics of the client using hardware equipment. Traditional password authentication systems can be stolen, lost or forgotten. To overcome these problems biometric play a vital role as a solution [5].

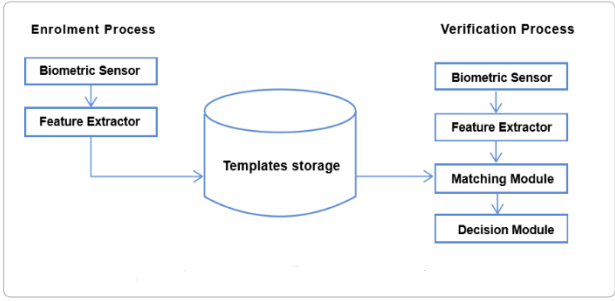

Fig. 5. The Mechanism for Biometrics Authentication [9]

Fig 5. illustrates that biometrics involves two steps while authenticating recipient, one is enrollment other is verification. Placing a finger on the device from multiple directions can make an issue if the server has only one sample. Therefore, when the client makes registration on the system with biometrics, it is recommended to record multiple fingerprint samples with measured characteristics in a digitalized format in the database.

For processing login method, registered users may place a finger on biometric sensor device then fingerprint characteristics are captured, extracted and processed towards Matching Module, which compares given values with stored samples from the database. If the new value is matched with the old sample, then the user will be authenticated, and access will be granted by the server.

Biometric has increased security in systems by preventing duplication of users, this process decreased the ratio of fake profiles in applications. Security PIN in biometrics is a physical characteristic of the user, so it has less possibility of losing the key.

Biometric authentication includes hardware devices for scanning fingerprint patterns, using hardware equipment with the application will increase the cost.

4.6 Open Authentication (O-Auth)

OAuth is a widely used protocol and open standard authentication system in modern applications. It is a mechanism that facilitates the recipient to use his identification from the third-party server. The registration process in application is escaped, if OAuth is performed. Most popular applications Facebook, Google, Twitter have implemented their APIs using OAuth protocol [11]. They provide services as OAuth APIs which allows vendors of different applications to use built-in methods for authentication on their systems.

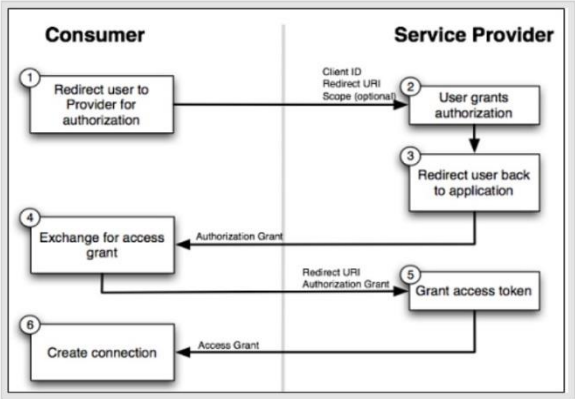

Fig 6. O-Auth Mechanism [12]

Fig 6. Illustrates steps for the O-Auth mechanism. In step 1 when a client enrolls the registration process on the application, it redirects to OAuth service provider like; Google, Facebook. In step 2 service provider authenticates the user with a request token and forwards him to the next step. Step 3, the user is redirected to application with the token having protected information, which can be used as query parameters. In step 4 user had an access token from the service provider and in step 5 user gets access to application resources.

In OAuth tokens are not expired automatically, because it has no defined time of expiration. It can be reused until terminated by the user [12].
Adoption of OAuth in applications escaped the registration process. It saved expenses of vender and time for the registration process and allows one account to be used at multiple applications instead of making individual registration for each. Some business applications have protocols to ask user secrete information like credit card number, trusting unknown server is a kind of risk for the user. If an application is performing OAuth with a client known server, it solves that problem and the user can provide secrete information on a trusted network. OAuth is standard protocol and it is secure to use because it uses SSL (Secure Socket Layer) for data transmission.

In current scenario users often use OAuth. Being careless on untrusted networks can be the reason for leaking private information, the client must aware of the application in which he is providing valid information. Phishing is considered as a disadvantage of OAuth, in which hackers provide fake graphical-user-interface (GUI) to the user and ask secure information like username and password which can be stolen if fake GUI is provided to the end-user.

4.7 Proposed Authentication Method MO-Auth

This study was carried out to gain knowledge about different authentication methods. Password-based authentication, Token based authentication and OAuth were adopted to design the MO-Auth authentication method. An API for MO-Auth authentication is designed and placed at the CMS based system to perform user authentication.

CMS system is assumed as a service provider on which we placed designed API to perform MO-Auth. On the other hand, a pre-designed web-based blog application was considered on which MO-Auth is used for authentication purposes. Which enabled clients of blog application to use credentials from third-party server i.e CMS system for authentication purpose.

Before performing the authentication process, it is necessary for the user to have a registered account on the third-party server. Like Google, and Facebook are providing O-Auth service for authentication

which enabled clients to process authentication in the different applications using credentials from Google or Facebook.
In our experimental work CMS system is considered as a third-party server, which provides MO-Auth Service. Initially, recipients must make registration in the CMS system, after registration users can use multiple applications associated with it. Like recipients uses credentials from Google or Facebook for login into a different system in which their APIs are configured.

After having an account on trusted server users can access services from vendor applications just by making one click on the button which will let the system perform MO-Auth for authentication.

Moreover, to configure our designed method MO-Auth we took API code from CMS system and configured it on our pre-designed Blog application to perform MO-Auth. MO-Auth is our designed methodology used for user authentication. It is a combination of three widely used authentication methods, Password, Time-based Token, and OAuth.

MO-Auth = Password-Authentication ∪ Time-based Token ∪ OAuth
Features of the above three authentication methods are coupled to enhance the security of important credentials over the web. MO-Auth is a chunk of code provided by the service provider, which is later configured in the vendor's application. After configuration whenever the client wants to access the system it eliminates the registration process and asks for sign-in form third party server (Service Provider).

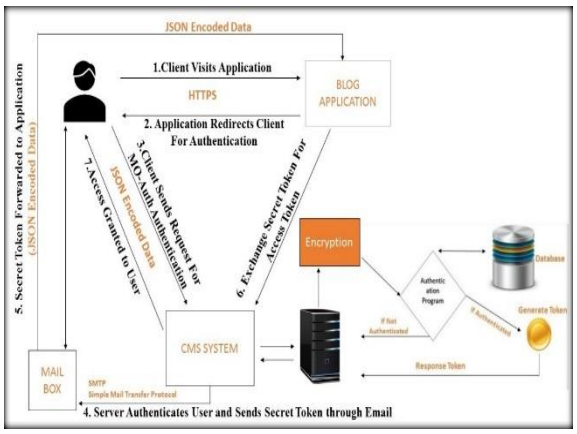

Fig 7. MO-Auth Mechanism

Fig 7 illustrates steps to perform MO-Auth. To validate our methods, an experiment was done by implementing MO-Auth in a CMS System which works as a service provider and configured it into another web-based Blog application as illustrated in figure 7. Steps for MO-Auth mechanism are defined as the following:

Step 1: Client visits an application, but services of application cannot be availed until the user makes registration or relate trusted server is used for authentication.
Step 2: Application asks for authentication and tells the user to be logged in with your trusted server.
Step 3: The client sends the request to his trusted server like we have mentioned our designed CMS System.
Step 4: Server of CMS System encrypts the password given by the user and executes the authentication process by comparing given values with database values. If a user is authenticated it generates a token and that token is forwarded to the user through email using SMTP (Simple Mail Transfer Protocol). If cookies are maintained already then, token will be generated automatically. The user has no need to provide the username & password.
Step 5: In step 5 user checks his mailbox and use received token, which redirects him towards application along with JSON encoded data.
Step 6: Application receives, decode and verifies JSON data. After that, it generates an access token and forwards it towards CMS-System.
Step 7: In Step 7 All processes are accomplished, the user will be authenticated and redirected to the application along with access token and identity attributes which will let him avail services.

Generated Sessions and Cookies are already maintained when a username and password are provided to the trusted servers at GUI. If the session and cookies are already maintained on a trusted server, MO-Auth skips step for password authentication and process authentication through one click. In the next step, the time-based token is generated, which is forwarded to the user through email and the user uses that token to get access over application.

Verified credential in JSON format is received as parameters of API. Application decode, and use received JSON data and verified access token to create, maintain new sessions and cookies, through which access control of the application is managed. Ajax, J-Query is used to send and receive user credentials encoded in JSON format and HTTPS protocol has been used as a transmission channel.

The designed authentication method can be used in any application. The vendor just needs to make registration in the CMS system and copy API code to configure it on the application as it is configured in the blog application.

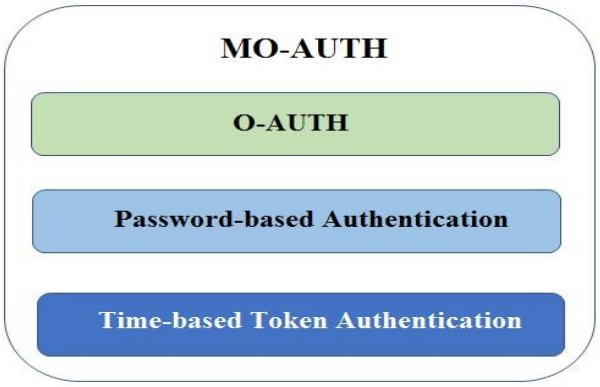

Fig 8. MO-Auth Layers

As we have already discussed MO-Auth is the combination of three authentication methods, Figure 8 illustrates layers in MO-Auth. O-Auth layer facilitates the client to use credentials from a third-party server, password Authentication layer verifies user credentials and time-based token generates token and sends it through email for verification. MO-Auth is based on three security layers, which makes the application more secure as compare to others. The user provides personal contact information while making registration on the CMS system, MO-Auth uses that information as security prospectus. Generated time-based access token is sent to the user by email through SMTP (Simple Mail Transfer Protocol) for confirmation and completes the authentication process. A generated access token is designed with time specification, if the user is not using the given token within defined time, it will be expired and will force the user to repeat the authentication process. MO-Auth has reduced registration cost, password maintenance cost, data processing cost, and cost for taking care of redundant data. And provides additional security by generating a time-based token for user verification through which phishing is prevented.

MO-Auth added an additional step in the sign-in process, so it consumes a little bit more time for a single login as compare to the password authentication method but adds more security to user credentials.

5. RESULTS

To validate our results, in this section we have discussed the survey we did on existing popular applications based on the availability of features as illustrated in table 2 and figure 9. We have found that no matter several novel approaches are being introduced but they did no impact on traditional methods. Traditional methods are still being used with the collaboration of novel approaches. Moreover, in this work, we have studied and discussed different authentication methods and proposed novel authentication method MO-Auth by combining three, password-based, time-based token and OAuth. We did an experiment by implementing designed API for MO-Auth in web-based application CMS system and Blog application. We considered a CMS system as a service provider and Blog application as an external system. MO-Auth was configured in Blog application and enabled the client to use user credentials from the CMS System for authentication.

TABLE II
FEATURE'S AVAILABILITY IN MODERN APPLICATIONS

Applications	Sign-Up	Sign-In Using OAuth	Sign-In Using Biometrics	Sign-In Using Token	Sign-In Using Password
Facebook	Yes	No	No	Yes	Yes

Gmail	Yes	Yes	No	Yes	Yes
Yahoo	Yes	No	No	Yes	Yes
Tweeter	Yes	No	No	Yes	Yes
WeChat	Yes	No	Yes	Yes	Yes
Instagram	Yes	Yes	No	Yes	Yes
Amazon	Yes	No	No	Yes	Yes

Table 2. illustrates the availability of features in modern applications. We considered seven famous applications and found that Sign-Up, Token and Password features are available in all of the applications, while OAuth is available in two of them and biometrics in one. In the current scenario, OAuth and Biometrics are modern methods to use in applications.

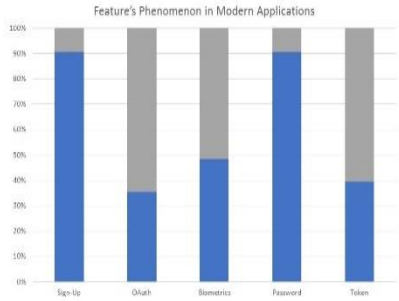

Fig. 9 Availability of Features in applications

Fig 9. illustrates the availability of features in the modern application as a graph. In the current climate according to the above graph, 90 % of applications have Sign-up (registration) and Password-based authentication. About 48% of applications are using biometrics for security, 38% uses Token-based authentication and 35% for OAuth. Sign up mechanism, Password and Token-based authentications are traditional methods are being used for authentication purpose, their usage in applications is more as compare to OAuth and biometrics.

The trend of biometrics using fingerprints is also spreading nowadays, some of these applications, especially Facebook can consider it as an additional feature for application, which can resist the duplication of users. The ratio of fake profiles will be decreased. OAuth is a modern standard nowadays used by famous applications. Many popular applications have already adopted the OAuth method, applications are migrating towards it for saving expenses which registration process takes. Both OAuth and Biometrics are new to applications, maybe in future percentage for their usage will increase.

TABLE III
COMPARISON OF ISO/IEC 9126 SOFTWARE CHARACTERISTICS BETWEEN AUTHENTICATION METHODS

Characteristics	Password-based Authentication	Token-based Authentication	Authentication Using Biometrics	O-Auth	MO-Auth
Usability	Uses String, numbers or special characters for authentication	Uses token for authentication	Uses physical characteristics of users like Fingerprint, Voice, Face, or Eye.	Uses User credentials from Third-party server	Uses dynamic time-based token.
Functionality	Uses traditional mechanism for interaction with the server	It includes an additional step of using token in password mechanism	It has two steps for authentication, enrollment, and verification	O-Auth is a combination of password and token authentication methods	It is a combination of three methods. Password, Token & OAuth
Reliability	Reliable if given password is strong and unpredictable	Stores cryptographic hash of the password, if the token compromised password is still protected	Cannot be replaced & cannot be used if the characteristic is lost or injured	It is reliable if the third party server is trusted	It is more reliable and secure to use because it enhanced an additional layer for security
Portability	The security key is memorized by the user.	The generated token is based password. The security key is memorized by the user	The Physical Characteristic of the user is used as a security key. The security key is always with the user	The security key is memorized by the user. The user receives the token from the server for sign-in	The security key is memorized by the user. Each time user attempt signs in, he receives new token via- email
Efficiency	Make direct interaction with the server	Uses two security layers for authentication	Fewer chances for losing the password. Its increased security, enrollment of invalid user is detected.	Uses two layers for authentication and facilitate the client to use credentials from the third-party server	Uses three security layers for authentication n. And facilitate user to use third party credentials & prevent Phishing
Maintainability	Low cost and convenient. Easy to maintain	Easy to maintain as a password, but it includes extra database columns in a table.	Uses hardware equipment's for authentication. Maintainability of biometric includes more costly as compared to other	Uses resources from a third party server. Low cost & easy to maintain as compared to Password, Token, and Biometrics	Low cost and easy to maintain as O-Auth

ISO/IEC 9126 is an international standard for evaluation of software quality characteristics [13], in Table 3 ISO/IEC 9126 Quality Characteristics between authentication methods are evaluated.

According to Table 3 result indicates that MO-Auth is easy to use, reliable, efficient, less cost, easy to maintain and much secure as compared to other authentication methods. Designed authentication method MO-Auth was examined through the human test, we considered 5 distinct users to use MO-Auth on applications on local server and found some results which are summarized among 10 based sample space, illustrated in table 4 as follows:

TABLE IV
REAL-TIME EVALUATION OF QUALITY FRAMEWORKS

User Attempts	Usability	Reliability	Efficiency	Cost	Maintainability	Security
1st User	10	9	9	10	10	9
2nd User	8	9	9	9	10	9
3rd User	9	8	9	10	10	9
4th User	10	9	9	8	10	9
5th User	8	7	9	9	10	9

Graphical summary of table 4 is illustrated as follows:

Fig 10. Real-Time Evaluation

Results illustrated in table 4 and figure 10 were collected at real-time testing, according to results frameworks are scored among 5 users and found usability as 45/50, reliability as 42/50, efficiency as 45/50, cost-saving 47/50 and security as 45/50. According to collected results, the designed method MO-Auth is found easy to use, reliable, efficient, less cost, easy to maintain and secure.

TABLE V
REAL-TIME EVALUATION OF QUALITY FRAMEWORKS

MO-Auth Units	Failed	Succeeded
Data Validation	No	Yes
Request	No	Yes
Response	No	Yes

Authentication	No	Yes
Tokens	No	Yes
Access Control	No	Yes

Furthermore, units of designed authentication were tested through POSTMAN extension and results were found as illustrated above in Table 5. It indicates each unit was working properly.

6. CONCLUSIONS AND FUTURE WORKS

This study was carried out to gain knowledge about authentication security in modern application development. We studied and discussed four distinct widely used authentication systems, including password-based authentication, token-based authentication, biometrics and OAuth along with advantages and drawbacks. Each method has its own mechanism advantages and disadvantages, we considered 3 of them to practice. So, we practiced and combined password-based authentication, token-based authentication and OAuth to propose a novel hybrid method called MO-Auth. To validate results, experimental work was done on two distinct web-based applications. One of these was considered as a service provider and other "Blog application" as an external application, which uses MO-Auth service to authenticate clients from CMS System. As a result, existing authentication methods and proposed MO-Auth were evaluated in the light of ISO/IEC 9126 software quality characteristics. Moreover, the survey was done on features available in popular web systems, survey result indicated usage of available features in modern applications. Moreover, proposed MO-Auth is much secure, less cost and easy to use in web-based applications.

Designed authentication method MO-Auth with two web-based applications were tested using the Apache server. Our goal is to facilitate web clients with proposed MO-Auth, in the future, we will test it at the remote server and make it available for over a web.

Conflicts of Interest: I declare that none of the authors have a conflict of interest.

7. REFERENCES

[1] McDaniel, P. (September 18, 2006). "Authentication." Pennsylvania State University, Information Networks, Springer. Series: Advances in Information Security.
[2] Kuldipsinh Jam et al, K. J. e. (3 March 2007). "Survey of Authentication and Authorization based on OAuth, M.Tech." International Journal of Innovative Research in Computer and Communication Engineering Volume 5 (Issue 3).
[3] Michal Trnka et al (June 12, 2018). "Survey of Authentication and Authorization for the Internet of Things." Security and Communication Networks Volume 2018: 17.
[4] Unnati Awasthi, U. (June 2017). "Token Based Authentication Using Hash Key, Session and Java-Mail API." International Journal of Innovative Research in Computer and Communication Engineering Volume 5 (Issue 6).
[5] Wencheng Yang et al, W. Y. e. (January 2019). "Security and Accuracy of Fingerprint-Based Biometrics." Symmetry Volume 11 (Issue 2).
[6] Homin K. Lee et al, H. K. L. e. (2007). "Cryptographic Strength of SSL/TLS Servers Current and Recent Practices."
[7] Krawczyk, H. (2001). "The Order of Encryption and Authentication for Protecting Communications." Springer.
[8] Martin.Drašar, M. (2009)."Password-based authentication." Available at: https://is.muni.cz/th/98998/fi_m/.
[9] 2019. "Token Based Authentication. Available at: https://www.vuemastery.com/courses/token-based-authentication/intro-to-authentication/.
[10] Wong KS, K. M. (2012). "Towards Biometric-Based Authentication for Cloud Computing." In the 2nd International Conference on Cloud Computing and Services Science.
[11] Feng Yang and Sathiamoorthy Manoharan, F. Y. a. S. (2013). "A Security Analysis of the OAuth Protocol." IEEE Pacific Rim Conference on Communications, Computers and Signal Processing.
[12] Chirang Solanki, C. (2017). "OAuth Security Overview." In Blog.
[13] British Standard (2011). "Systems and software engineering, System and Software Quality Requirements and Evaluation (SQuaRE) – System and software quality model." BSI Standard Publication.

