I J C T A, 9(11) 2016, pp. 5253-5262

MOBILE BASED SECURE AUTHENTICATION USING TLS AND OFFLINE OTP

Vishal Gangwar[*], Ravishanker[**] , and Dr. Ashish Kr. Luhach [***]

Abstract: World Wide Web hosts (e.g. Yahoo, Gmail, etc.) deploy the best known security mechanisms to protect user data from attackers; still the personal information is compromised. Security lapse is at the user end, i.e. user's personal faults are responsible for the onslaughts. Assuming access to web servers is really much difficult for hackers, so they attempt to advance access to user's system to steal data. In this paper, an enhanced authentication scheme is proposed to maintain confidentiality of information. The proposed framework will utilize a pre-shared phone number and MAC address of the device along with the current timestamp (PMT), required to generate TOTP (Time-Based One time Password) in order to generatet an offline secret hash code using offline token generation mobile app. The generated hash code is entered by the user on the website and transferred to the server using TLS (Transport Layer Security) connection established between server and user system. It also does away with the usage of SMS based OTP applications which are strung-out on the cellular net.

Key Words: PMT, SSL/TLS, multi-factor authentication, web security, OTP

1. INTRODUCTION

Data sharing and storing over the cloud network using internet becoming a movement. Need for securing the data over the internet is increasing. Use of smart phone has increased. People like to exchange data over the internet using smart phones. Smartphone based web applications are developed to provide ease to the users. Hacking the web application servers is also complicated as compared to getting access to the user system in parliamentary procedure to steal data. Hence, attacks normally happen at the user terminal. The major goal of net security is to prevent unauthorized access to data and resources. Various cryptographic techniques are applied by clients and servers to keep the confidentiality of data [1].

Authentication is the heart of every security model. It is the process to confirm the user's identity (or a machine), attempting to gain access to a system or resource. Password based authentication is the most often utilized and trusted authentication mechanism. User needs to insert the required login credentials (username and password), to acquire access to a resource or computer, the supplied credentials are then matched against a database which contains the list of all authorized users and their passwords. Many advances have been suggested for proper strategies of securing and using passwords [1][2]. The user is suggested to maintain strong passwords, however number of problems persists in password based authentication mechanism.

[*] Department of Computer Science and Engineering, Lovely Professional University, Punjab, India
vishu.gangwar101@gmail.com

[**] Department of Computer Science and Engineering, Lovely Professional University, Punjab, India
ravishanker20@gmail.com

[***] Department of Computer Science and Engineering, Lovely Professional University, Punjab, India
ashishluhach@acm.org

A better technique which overcomes the shortcomings of password authentication technique is known as multi-factor authentication. Multi-factor technique is considered to be much secure as it adds up extra layer(s) of protection over password authentication technique. It employs two or more universally recognized authentication factors: a knowledge component (“something user needs to remember e.g. PIN, password”), a possessing factor ("something user possess, e.g. tokens, smart cards"), inherence factor (“elements related with user e.g. biometrics”) [3].

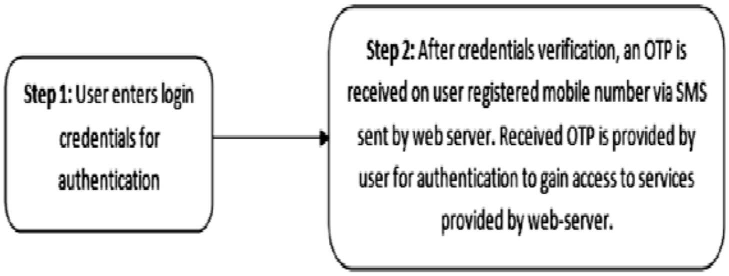
Step 1: User enters login credentials for authentication
Step 2: After credentials verification, an OTP is received on user registered mobile number via SMS sent by web server. Received OTP is provided by user for authentication to gain access to services provided by web-server.

Figure 1: Two-Factor Authentication

2. OVERVIEW OF EXISTING METHODOLOGIES FOR AUTHENTICATION

In the literature; approaches related to multi-factor authentication have been introduced. Two-factor authentication proposed in [4], the knowledge component is the email and the pin is sent to the registered electronic mailwith the corresponding user account. An authentication scheme proposed in [5] is using biometrics as one of the key elements for authentication. Granting to the suggested approach in [6] the password is represented using a graphical icon, which is beamed from the service providers to the user on the mobile device and the user has to point appropriate points to insert the word.

In authentication based on SMS, a random value is generated which is termed as OTP, is sent to the user registered mobile number by service providers through an SMS [7], which is then submitted by the user for authentication; this is a widely used multi-factor authentication scheme. [8] and [9] proposed generating location based OTPs and encryption schemes. Another suggested approach for multi-factor authentication is using SecureID [10]. It utilizes a security device which generates an OTP which is then provided by the user along with other credentials for authentication. Each device uses a unique seed to generate OTP and the seed is also stored in the server’s database in order to validate the OTP sent by the user. The device is timely synchronized with the host. Although the technique holds the benefit over SMS based authentication as it is free and worldwide accessible, but it will be really costly and infeasible for free service providers like FACEBOOK to deliver a security device to each customer.

The OTPs can also be generated by sharing the seed between both the communicating parties i.e. client and host. On every successful login the seed is regenerated. Use of IMEI and IMSI numbers as the seed to generate OTPs is proposed in [11]. Simply as this information is known to the mobile service provider, it is not considered safe to practice. Time based OTP generation is proposed in [12] but this technique is most usually employed for generating SMS based OTPs. Usage of TLS to exchange the seed for the generation of TOTP is proposed in [13]

If the OTP is sent to email [4] of the user and the intruder hijacked the user email account, then he can easily generate and use OTPs to access the services pretending to be an authentic user if the intruder knows the user login credentials as well.

This report gives an authentication framework which is more secure and trusted as compared to other systems. It utilizes the device hardware address, i.e. MAC address (something user possesses like a smart card) and number (pre-shared with the server) along with the current time (PMT) to authenticate the user. Compared to SMS based authentication schemes the technique uniquely authenticates the user using pre-shared MAC and number which helps in eliminating the dependency on other networks (like cellular networks in case of SMS based OTP scheme) to send the OTP. Hence, extra cost for sending OTP through SMS using additional network is reduced. If in some way the intruder gets physical access to the user registered number for SMS based OTP services, then he can easily get the OTP through SMS in SMS based approach. If the intruder intercepts the OTP message, then he may use the OTP to access the services before the user does. PMT mitigates the defects of SMS based multi-factor authentication system.

SecureID[10] utilizes a separate device to generate an OTP based on the fixed seed known by the device as well as host, whereas PMT is based on the fixed device hardware address and modifiable pre-shared number. The additional device needed to generate the OTP increases the price and if the device is stolen by the intruder along with the user’s credentials, then he can misuse the services. The application is PIN protected so unauthorized users cannot generate the OTPs on acquiring the device.

A similar technique using the GPS location along with pre-shared number and timestamp is proposed in [14] to authenticate users. It will increase the communication overhead as each time the user logs in, the GPS coordinates need to be calculated and sent to the server side and the server fetch the user GPS coordinates information from GPS server and hence increasing the load on the server too. The GPS positioning of the user can easily be followed by the intruder as well as it will compromise the user’s “DO NOT TRACK” feature. Also the intruders can social-engineer the user to get the pre-shared number. The location based OTP generation schemes possess the same problem. All the location based techniques are based on GPS coordinates which requires internet connectivity to the mobile device as well in order to generate the OTP.

In PMT technique, the pre-shared number can be switched anytime, but the MAC address of the device needs to be fixed. The technique will mitigate all the shortcoming of the above listed techniques utilized to generate OTPs for authenticating the user. Further, the paper is organized in the following sections: Section 2 explains the detailed PMT authentication technique. Security analysis is presented in Section 3. Paper is concluded in Section 4.

3. PROPOSED FRAMEWORK

This section will describe the working of multi-factor authentication scheme PMT. The technique can substitute the existing SMS based authentication due to its security characteristics. Pre-shared number and hardware address helps in identifying the authenticity of the user and verification of the device used by the user. The device and number are registered with the server during the user registration phase. At the time of enrollment, a user is also commanded to select an image and a security question which is stored on the host. The number can be modified by the user anytime, but the hardware address is required to be identical. The multi-factor PMT authentication technique involves three phases: the first phase uses an older method of using username and password for authentication. In the second phase hash is calculated. Finally, in the third phase the authenticating party (server) verifies the hash sent to it by the PMT app.

The notations used to describe the proposed work area:

Table 1
Notations Description

Notation	*Description*
D_i	The device on which PMT app is installed by the user
W_S	Web-Server
T_S	Token Server
U_{ID}	User unique id registered with the server
P_W	User password registered with U_{ID}
IT_U	Identity Token generated at User side
IT_S	Identity Token generated at Server side
T_{OTP}	Time Based OTP
MAC_U	Mac address of the D_i
P_{NO}	The number registered with W_S to corresponding U_{ID}
C_{RT}	Current Time
H	One way hash function
H_U	Hash (one-time identity token) generated by the application
H_S	Hash (one-time identity token) generated by T_S

Phase 1: Log-in using the password

In this phase, the user enters the web page URL of the website he needs to gain access using any internet enabled device. The requested webpage server sends the authentication page to the user on which he needs to enter the login credentials (U_{ID} and P_W). The hash of entered credentials is sent to W_S for verification. If the hash of credentials sent by user to W_S is successfully verified, then the user is asked to provide IT_U by W_S.

Phase 2: Generation of IT_U and ITS

In this phase, IT_U is generated by the PMT app installed on the D_i. The identity token IT_U is generated using an equation:

IT_U = hash (hash (Pre-shared number $\oplus$ Pre-shared MAC) $\oplus$ T_{OTP}); (1)

The T_{OTP} is automatically generated by the application using C_{RT} entered by the user which is displayed as a result of an ITU request by W_S. Concurrently the current time used to generate T_{OTP} is also sent to server by establishing a TLS connection between the user web accessing device and W_S. The seed used to generate TOTP is changed every time the user logs-in to PMT application and the seed needs to be renewed after every 7days.

At the same time the CRT, received by the W_S is sent to T_S for IT_S generation by using equation (1).

Phase 3: Verification of Identity Token

IT_U entered by the user is sent to W_S using TLS connection. T_S sends IT_S to W_S. If both IT_U and IT_S matches, the user successfully passed the authentication procedure and allowed to access the services offered by W_S.

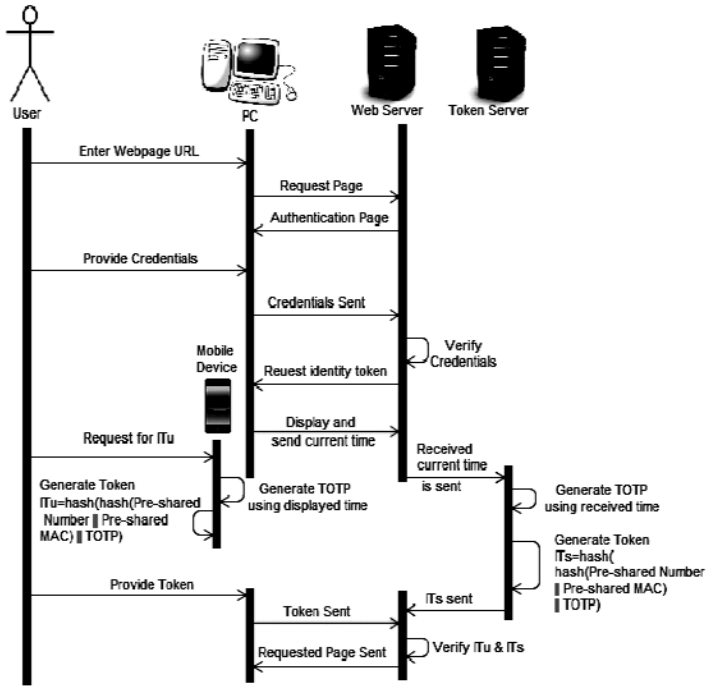
User
PC
Web Server
Token Server
Enter Webpage URL
Request Page
Authentication Page
Provide Credentials
Credentials Sent
Verify Credentials
Mobile Device
Reuest identity token
Display and send current time
Request for ITu
Received current time is sent
Generate Token ITu=hash(hash(Pre-shared Number ‖ Pre-shared MAC) ‖ TOTP)
Generate TOTP using displayed time
Generate TOTP using received time
Generate Token ITs=hash(hash(Pre-shared Number ‖ Pre-shared MAC) ‖ TOTP)
Provide Token
Token Sent
ITs sent
Verify ITu & ITs
Requested Page Sent

Figure 2: Proposed Architecture of Framework

Algorithm 1: First time application registration with T_S installed on D_i

It is used to exchange user device information and seed required for the generation of TOTP, which is further required to generate an identity token. A TLS connection is established between client and server for secure information exchange. A user is required to log-in using tradition method, i.e. enters the username and password which are verified by the token server. Along with a user id and password hardware address of the device, i.e. MAC address is sent which is stored by token server. The token server generates a seed for generation of TOTP and stores it in the database and sends the seed to device too using TLS. The device asks the user for a registered phone number with web server and PIN required for accessing the application. Application computes a 32-bit one way hash using the phone number and MAC address and stores the seed and hash in encrypted form.

Step 1: User needs to login into the application using required credentials.

Step 2: TLS connection is established between the application and $T_{S.}$ A confirm the TLS connection and the session key is sent to an application by $T_{S.}$

Step 3: A seed generation request is generated and send to T_S by application. The parameters of the request, i.e. login credentials along with MAC_U are encrypted and send to T_S for verification.

$$Encrypt\ [data = H\ (U_{ID} \oplus P_W);\ MAC_U;\ cipher = AES256CBC]$$

Step 4: The server decrypts the data.

Decrypt [key=session key; cipher=AES256CBC; data=seed request]

Step 5: T_S verifies the credentials of the user, stores the MAC_U in the database corresponding to U_{ID} and a unique 32-bit seed is generated for the user account.

Step 6: The server uses the same TLS connection to send seed to the application.

Encrypt [key = session key; cipher = AES256CBC; data = seed]

Step 7: PMT application decrypts the seed

Decrypt [key = session key; cipher=AES256CBC; data = encryptedseed]

Step 8: Application requires the user to enter P_{NO} registered with the server and a PIN for opening the application to prevent unauthorized use.

Step 9: Application generates a 32-bit hash string using MAC_U and P_{NO}.

$$H_U = H\ (MAC_U \oplus P_{NO})$$

Step 10: Application encrypts H_U and seed and stores them.

After successful registration of the device with token server, the application is ready to generate one time identity token offline i.e. without using the mobile network or internet connectivity. The user is required to update the seed with the token server after every week. The TLS session required to exchange the information is valid only for some time (e.g. 5 minutes).

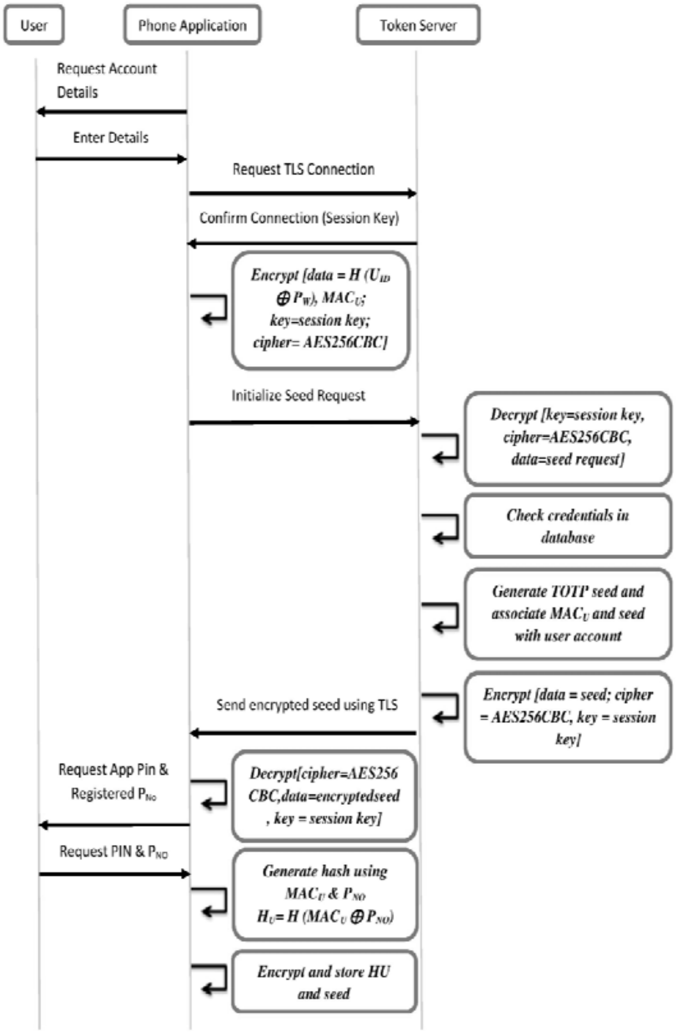
User
Phone Application
Token Server
Request Account Details
Enter Details
Request TLS Connection
Confirm Connection (Session Key)
Encrypt [data = H (U_{ID} $\oplus$ P_W), MAC_U; key=session key; cipher= AES256CBC]
Initialize Seed Request
Decrypt [key=session key, cipher=AES256CBC, data=seed request]
Check credentials in database
Generate TOTP seed and associate MAC_U and seed with user account
Encrypt [data = seed; cipher = AES256CBC, key = session key]
Send encrypted seed using TLS
Decrypt[cipher=AES256 CBC,data=encryptedseed , key = session key]
Request App Pin & Registered P_{No}
Request PIN & P_{NO}
Generate hash using MAC_U & P_{NO} H_U= H (MAC_U $\oplus$ P_{NO})
Encrypt and store HU and seed

Figure 3: Exchange of seed and MAC_U b/w T_S and Application

Algorithm 2: Generation of OTP offline at client side

After successful registration of the application with the token server, the application is ready for generating one time identity tokens. To generate a token user needs to enter the PIN required for opening the application and request the application to generate identity token. The user is then asked for the current time displayed on the PC when the web server requests PC to provide identity token. When the user provides the current time displayed in the application, it decrypts the seed and hash of phone number and MAC address. Using seed and current time application generates a 32-bit TOTP and computes a 32-bit identity token, i.e. hash of user MAC and phone number hash and TOTP.

Step 1: User opens the application in D_i and enters a PIN to gain access to application services.

Step 2: User requests the application for IT_U generation.

Step 3: User needs to enter the C_{RT} displayed on the device using which the user is communicating with W_S in the application.

Step 4: Application decrypts the seed and H_U.

Step 5: Using seed and provided C_{RT} a 32-bit T_{OTP} (random number) is generated.

Step 6: The application generates a 32-bit token IT_U by using H_U and T_{OTP}.

$$IT_U = H\,(H_U \oplus T_{OTP})$$

After completing the steps identity token is successfully generated and displayed to the user. The identity token is valid only for a particular period of time. The user needs to provide this identity token to the web server by entering it in PC when asked.

Algorithm 3: Generation of OTP at server side

When the user registers the application first time with token server, it stores the seed generated by it and sent to application in database and ask the web server to provide the registered phone number of the corresponding user and computes a one way 32-bit hash using MAC and phone number and stores it too in the database. When the web server requests the user for identity token current time is sent by the PC to the web server using TLS connection. The current time is then sent to token server; it generates a 32-bit TOTP using seed stored in the database and the current time provided by the web server. The identity token for the user is generated by computing one-way 32-bit hash of generating TOTP and hash of user MAC and phone number stored in a database of token server registered with the corresponding user.

Step 1: During first time registration of application with T_S, it stores the MAC_U of D_i acquired from application and the generated seed associated with U_{ID} in the database.

Step 2: After seed generation T_S asks the W_S for P_{NO} registered with the U_{ID}.

Step 3: T_S generates a 32-bit hash string using MAC_U and P_{NO} and store it in a database.

$$H_S = H\,(MAC_U \oplus P_{NO})$$

Step 4: When a user requires using the services provided by W_S, an identity token request is sent by W_S to the user. The user device asking to access the service displays and transmits C_{RT} to W_S using TLS.

Encrypt [data = C_{RT}; cipher = AES256CBC; key = session key]

Step 5: W_S decrypts C_{RT} and transmits the decrypted C_{RT} to T_S.

Decrypt [cipher=AES256CBC; data = EncryptedC_{RT}; key = session key]

Step 6: T_S extract seed and H_S from the database associated with the U_{ID}.

Step 7: Using seed and provided C_{RT} a 32-bir T_{OTP} (random number) is generated.

Step 8: Then T_S generates a 32-bit token IT_S by using H_S and T_{OTP}.

$$IDs=H\ (H_S \oplus T_{OTP})$$

After these steps identity token for the user is successfully generated at the token server without communicating directly with the client and thus ensuring security as the requests for token generation comes through the web server. The token server only communicates with the application and web server directly.

Algorithm 4: Verification of identity token

After the identity token is successfully generated for the user at both ends user and server; the user needs to provide the generated token to the web server which is being sent to the web server using TLS connection and the token server also send the generated identity token for the corresponding user to the web server. The web server matches both the tokens provided by the user and token server.

Step 1: User enters the IT_U generated by the application.

Step 2: User transmits IT_U using TLS to W_S.

Encrypt [data = IT_U; cipher = AES256CBC; key = session key]

Step 3: W_S decrypts C_{TR} and transmits the decrypted C_{TR} to T_S.

Decrypt [cipher=AES256CBC; data = EncryptedIT_U; key = session key]

Step 4: W_S requests T_S to send IT_S.

Step 5: If IT_U=IT_S, the UID is allowed to access services provided by W_S else denied.

If both the token matches, then user is treated as authentic user and allowed to access the web services provided by the web server else the services are denied to the user and the session with the user is terminated.

4. SECURITY ANALYSIS OF PROPOSED FRAMEWORK

The possible threats and how they can be neutralized using the PMT multifactor scheme are discussed in this section.

Scenario 1: The intruder gains access to user login credentials by any means. In this case, the intruder cannot login to use the services due to pre shared number and MAC of the device used for generating the identity token.

Scenario 2: The attacker knows the shared number along with user login credentials, still the intruder cannot gain access to services as the MAC address used to generate identity token is unique for every device. Hence, the generated identity token of the intruder will not match with that of the server.

Scenario 3: Somehow the intruder intercepts the token sent to the server along with user credentials; still he cannot use the token later for authentication due to limited lifetime period and the generated token can only be used once.

Scenario 4: Intruder intercepts the current time being sent to the server and knows the pre-shared number along with user credentials and somehow generates a valid T_{OTP} too; still cannot access the web service due to limitation of hardware address i.e. MAC (the unique address for each device) to generate the token.

Scenario 5: The app is PIN protected, so that if the intruder is able to get physical access to the user device, he will not be able to generate identity token in order to access the services.

Scenario 6: In case the user wants to access the services on any other device or the device is lost, then the user can gain access or register the new device by following several steps required to authenticate the user:

Step 1: Log-in using username and password.

Step 2: Select whether to send the OTP to email address or the backup phone registered with W_S during the registration phase.

Step 3: Enter the received OTP, select the image and answer your security question registered with W_S during the registration phase.

Step 4: To register new mobile device, select change the existing device information with the current device information in case of new D_i, else the services provided by W_S will be accessible by the user in case of PC.

All these steps are executed by establishing a TLS connection between node and host. After sticking with these steps the user can access the services provided. A user is required to install PMT app on the new device as the user can access the services using the above mentioned steps only three times a day without the PGT app.

E-commerce is widely used to commit transactions over internet. There has been a significant growth in e-commerce industry. Due to easy availability of internet and computer systems, trend of committing transactions over internet is increasing. The easy availability has created a number of trained intruders to harm the e-commerce industry for their personal benefits. [15][16] Provide methods to secure the e-commerce websites and transactions committed using them. Incrementing the offline OTP along with the provided security standards will enhance the e-commerce security and prevent the intruders from gaining unauthorized access to data and thus helps in maintaining confidentiality of data.

5. CONCLUSION

An efficient network independent mobile based authentication scheme is proposed in this paper. The framework uses the pre-shared number and MAC address of the device along with TOTP to generate a hash known as the one-time identity token to successfully authenticate the user attempting to access the services offered by the network host. Unlike SMS and location based multi-factor authentication schemes, it does not require network services to transmit or to generate the OTP for authenticating the user. It never transmits the pre-shared number and the MAC address of the device during the token generation process. MAC is only shared once through the channel at the time of application registration on token server, thus making the intruder difficult to guess and the number can also be modified by the user. The technique can easily be implemented in a vast number of applications involving multi-factor authentication security. In future, the factors required to identify and authenticate the device and user can be improved to enhance the security level.

References

[1] B. Ross, C. Jackson, N. Miyake, D. Boneh, J.C. Mitchell, "Stronger password authentication using browser extensions." Proceedings of the 14th Usenix Security Symposium. 2005.

[2] L. Lamport, “Password authentication with insecure communication.” Communications of the ACM, pp. 770-772. 1981.

[3] Bauckman, Dena, Terry, Nigel, Paul, Johnson, David, Joseph, Robertson, “Multi-Factor Authentication.” U.S. Patent No. 20,130,055,368. 28 Feb. 2013.

[4] Weber, Frank, “Multi-factor authentication.” U.S. Patent No.7, 770, 002, 3 Aug. 2010.

[5] S. Patil, K. Bhagat, S. Bhosale, M. Deshmukh, “Intensification of security in 2-factor biometric authentication system.” Pervasive Computing (ICPC), International Conference, pp. 1-4, IEEE. 2015.

[6] A.P. Sabzevar, A. Stavrou, "Universal multi-factor authentication using graphical passwords," Signal Image Technology and Internet Based Systems. SITIS'08. IEEE International Conference, pp. 625-632, IEEE. 2008.

[7] S. Indu, T.N. Sathya, V.S. Kumar, “A stand-alone and SMS-based approach for authentication using mobile phone.” In Information Communication and Embedded Systems (ICICES), International Conference, pp. 140-145, IEEE. 2013.

[8] W.B. Hsieh., J.S. Leu, “Design of a time and location based One-Time Password authentication scheme.” In Wireless Communications and Mobile Computing Conference (IWCMC), 7th International, pp. 201-206, IEEE. 2011.

[9] S.V. Limkar, R.K. Jha, S. Pimpalkar, S. Darade, “Geo-Encryption-A New Direction to Secure Traditional SSL VPN.” In Information Technology: New Generations (ITNG), Eighth International Conference, pp. 1070-1071, IEEE. 2011.

[10] M. Nystrom, “The SecurID (r) SASL Mechanism.” Network Working Group Request for Comments, RFC, 2808. 2000.

[11] F. Aloul, S. Zahidi, W. El-Hajj, “Two factor authentication using mobile phones.” Computer Systems and Applications. AICCSA 2009. IEEE/ACS International Conference, pp. 641-644, IEEE. 2009.

[12] D. M’Raihi, S. Machani, M. Pei, J. Rydell, “Totp: Time-based one-time password algorithm.” Internet Requests for Comments, Internet Engineering Task Force (IETF), RFC, 6238. 2011.

[13] S. Singh, Ravishanker, “An improved network independent two-step authentication scheme using TLS.” International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.55, pp. 2261-2264. 2015.

[14] U.A. Abdurrahman, M. Kaiiali, J. Muhammad, “A new mobile-based multi-factor authentication scheme using pre-shared number, GPS location and time stamp.” In Electronics, Computer and Computation (ICECCO), International Conference, pp. 293-296, IEEE. 2013.

[15] A.K. Luhach, S.K. Dwivedi, and C.K. Jha, “Implementing the Logical Security Framework for E-Commerce Based on Service-Oriented Architecture.” In Proceedings of International Conference on ICT for Sustainable Development (pp. 1-13). Springer Singapore. 2016.

[16] A.K. Luhach, and R. Luhach, "Research and implementation of security framework for small and medium sized e-commerce based on soa." Journal of Theoretical and Applied Information Technology, 82(3), 2015.