International Journal of Research in Engineering, Science and Management
Volume-2, Issue-4, April-2019
www.ijresm.com | ISSN (Online): 2581-5792

Multi-Factor Authentication using Mobile Phones

Manisha Jadhav[1], Shital Dudhal[2], Suyog Mate[3], Arpita Raut[4]
[1,2,3]*Student, Department of Information Technology, Vidyalankar Institute of Technology, Mumbai, India*
[4]*Professor, Department of Information Technology, Vidyalankar Institute of Technology, Mumbai, India*

***Abstract*: Online and Enterprise resources typically required authentication before user allow to access the sensitive applications and information. Sensitive information contains user personal information, transactions, confidential data, etc. Traditional user authentication used user identifier, Password, Personal Identification Number (PIN), Token code etc. Existing Systems can't fulfil the current requirement of the user authentication. So that most of the system used multilevel and multifactor authentication mechanism to allow authorized user to get access the sensitive application and information. Recently such multifactor security is provided using Risk Based Authentication (RBA) mechanism. The RBA provides access based on Enforcement policy and access decision based on the risk score. Due to which RBA mechanism provides a more secure way to access the sensitive application and information by the user. In this project, we will propose RBA mechanism based on User's machine specific authentication information generate OTP using algorithm i.e. sha256 and md5.In our project during OTP generation no Network present that is more secured than other system.**

***Keywords*: Chain Hashing, Client, Hash Collision, MD5, OTP, Seed, Server, SHA-256.**

1. Introduction

Authentication is the process of verifying whether someone or something is, in fact, who or what it declares itself to be. Authentication is a part of everyday life in the digital age. It is a technology provides access control for systems by checking to see if a user's credentials match the credentials in a database of authorized users or in a data authentication server. The process is completed when authorized user is authorized for accessing system after matching the credentials. In case of authorization the admin or the super user does the job of granting rights and checks the permission of the user account for accessing the resources. The user's permissions are stored locally or on the server and define the privileges and preferences granted for the authorized account. The administrator sets the settings which are defined by environment variables. Existing system of authenticating and authorizing users is generally carried out using One-factor Authentication (1FA) or Two-factor Authentication (2FA). 1FA accepts 2 fields, an Email-ID/Phone Number/Username and a password. The latter is more secure than the former as it adds extra level of barrier for logging into websites. However, 2FA is carried out using SMS channel which is insecure and prone to attacks. In our case, Online banking requires strong user authentication. User authentication is often achieved by utilizing a one-factor or two-factor authentication technique based on something the user knows, i.e., a static password, and something the user has, i.e., an OTP. The traditional 2FA system works by sending an OTP over an SMS to a user who wants to make an online transaction.

2. Aim and Objective

The aim of this project is to improved level of security and delivers an authentication assurance for sensitive transactions. Our system overcomes the problems with utilizing OTPs with an SMS, consisting of the SMS cost and delay, along with international roaming restrictions and bypassing of OTP. Our idea is to produce multiple OTPs from an initial seed in a parallel process with the service provider itself, e.g., an online transaction, by utilizing two different types of hash functions, which come with a nested chain. The proposed scheme can resist an off-line guessing attack because it uses strong passwords produced from strong hash functions. Hence this project will discuss OTP production in the forward direction. This production will completely eliminate the mentioned limitations.

3. Problem statement

Whether a single layer of security is devised to authenticate the identity of customers or internal team members, the absence of additional protective measures can leave your stored data vulnerable to attack. Cyber terrorism attacks are growing ever more advanced and sophisticated, with security barriers providing limited protection when implemented on their lonesome. Password protection and other basic forms of identity authentication often provide simple pickings for cyber criminals, able to decrypt the less-than-secure codes. To overcome this problems, we will design and develop a wireless secure system.

4. Literature survey

A. One-time password

Leslie Lamport was the first to suggest the idea of OTP in the early 1980s [1]. He stated from his principle that each time a

user logs into the system, a pre-defined algorithm generates a pseudorandom output which increases the security. An OTP is valid only for a single login transaction or session.

B. The S/Key OTP System

The S/KEY one-time password authentication system uses a computation to generate a finite sequence of single-use passwords from a single secret "seed." The security is entirely based on this seed, which is known only to the user. The single-use passwords make it computationally intractable way to compute any password from the preceding sequence [2].

C. Bicakci et al.'s Scheme

The infinite length hash chains (ILHC) uses to produce a forward and infinite one-way function (OWF) [3]. This OWF is the OTP production core. Bicakci et al. proposed a protocol using RSA, where d is the private key and e is the public key. The OTP originating from initial input "s" using the RSA public-key algorithm for the t^{th} authentication is

$$Otp_t(s)=A^t(s,d)$$

And the verification of the t^{th} otp is done by decrypting $Otp_t(s)$ using e: $A(Otp_t(s),e)=Otp_{t-1}(s)$[6]

D. One-factor authentication (1FA)

One-factor authentication (1FA) gives a secure access to user in authentication system [4]. It uses only one category of credentials eg. Passwords or PIN code. for securing access to a given system, such as a network or website, that identifies the party requesting access through only one category of credentials. Eg. Passwords or PIN Code. The most common example of 1FA is password-based authentication. Best practices include creating a strong password and ensuring that no one can access it. One of the main troubles with passwords is that most users either don't understand how to make strong and memorable passwords or underestimate the need for security.

E. Two-factor authentication (2FA)

The user process 2FA authentication to verify themselves to better protect both the user's credentials and the resources the user can access but this security process done with two different authentication factors [5]. It is also Known also as multi factor authentication, 2FA requires two different means of verifying a user's identity when logging into a secure account. Typically, one factor is a physical token, such as an ID card, and the other is something memorized, such as a security code or password. In 2FA security hackers have a more difficult time breaking into an account even if they get their hands on the username and password

- This authentication preferably uses two of the following types:
- Something you know, like password, pattern or personal identification number (PIN).
- Something you have, like a key fob, RSA token or code sent via an SMS text.
- Something you are, such as a fingerprint, iris scan, voiceprint or even your face.

5. Proposed system

It is multifactor authentication system were user process some authentication steps to access the bank website, here one android app also included. User registers on the bank's website via mobile app because it is device base system. It we register via desktop is not possible in this system as we cannot provide IMEI & IMSI numbers directly inside the browser text fields. Registration page contains parameter such as username and email-id, password, mobile no., security question and corresponding answer. Using Android's permission, the mobile app reads and sends the IMEI and IMSI number to the server. Server receives all the input data and creates an account for the user with following parameters - unique username bound to the email-id, password set by user. Simultaneously it creates one unique bank ID and send to the user registered mobile no. for login credential to bank website. Server will concatenate IMEI, IMSI to generate a seed.

User will now request login into website using unique bank ID. Server will validate the input data and create a secure SSL session. Server will randomly generate and display 2 index variables (x, y) and send them to the client. The server will use the same variables to generate a server sided OTP from seed using nested hashing functions or a hash chain. User will input the server generated 2 index variables (x, y) in the mobile app's OTP Section. The mobile app will generate a client sided OTP using nested hashing functions from the seed. User will now enter the client sided OTP in the browser. Server will now compare its own generated OTP and client sided OTP. If they are same, the authentication proceeds to the third factor i.e. security question. User is prompted for a security question from the server. If the answer is successfully verified, the user is fully authenticated into the system.

A. Flow of the system

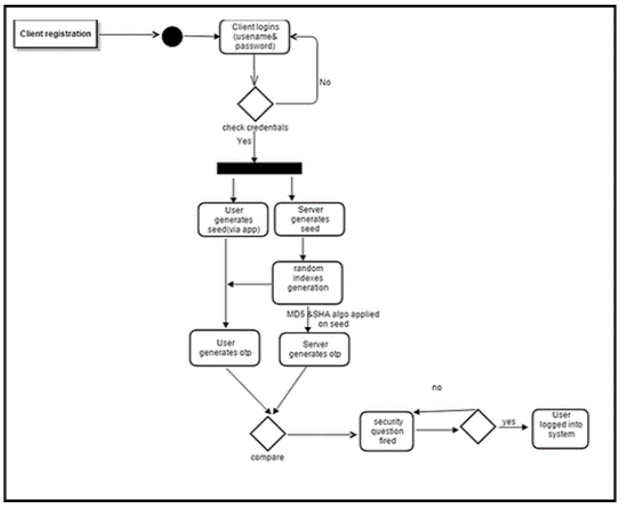

Fig. 1. Flow of the system

B. Hashing function

1) Message Digest 5 Algorithm (MD5)

Padding bits and Append Length: Padding of the bits is compulsory with '0' and '1' first and last respectively until the resulting ≠ bit length which is equal to 448 mod 512, and the last of bit length of the original message m_n as 64-bit integer. If last bit is padded the length of the message is 512N for a true integer N [6].

Divide the input message into 512-bit blocks: The message already padded is now partitioned into N successive 512-bit blocks m_1, m_2.....m_n.

Initialize Channing variables: Initialization of 32-bit number in the form of chaining variables (A, B, C, D) these values are represented in hash only

A = 01 17 2d 43
B = 89 AB CD EF
C = FE DC BA 98
D = 76 54 32 10

Process blocks: The four buffers (A, B, C and D) messages are joined now with the input words, using the four auxiliary functions (W, X, Y and Z).4 rounds are performed and each involves 16 basic operations. The Processing Block denoted as P which applied to the four buffers (A, B, C and D), by using message word M[i] and constant K[i]. The item "<<<s" denotes a binary left shift by s bits. The four type of info related functions that each take as input three 32-bit words and produce same bits of output i.e. 32-bit word. They apply the logical operators ^, v,! and XOR to the input bits.

Q (A, S, D) = AS v NOT (A) F
W (A, S, D) = AS v S NOT (F)
E (A, S, D) = A XOR S XOR F
R (A, S, D) = S XOR (A v NOT (F)

The bits of A, S, and D are totalitarian and balance each bit of Q (A, S, D) will be totalitarian and balance. The functions (A, S and D) is equal to P, in that they do job in "bitwise parallel" to produce the reliable output from the bits of A, S and D when the similar bits of D, E and F are autarchic and balanced, then each bit of W (A, S, D), E (A, S, D) and R (A, S, D) will be totalitarian.

2) Secure Hashing Algorithm (SHA)

- *Padding:* SHA adds Padding to the end of the genuine message length is 64-bits and multiple of 512[6].
- *Appending length:* Here excluding length is calculated.
- *Divide the Input into 512-bit blocks:* In this we divide the input in the 512 bit blocks
- *Initialize chaining variables:* Here we initializing chaining variables here we initialize 5 chaining variables of 32 bit each=160 bit of total.
- *Process Blocks:* Copy the chaining variables. Divide the 512 into 16 sub blocks. Process 4 rounds of 20 steps each

6. Scope

Our idea is to produce multiple OTPs from an initial seed in a parallel process with the service provider itself, e.g., an online bank, by utilizing two different types of hash functions, which come with a nested chain. The resulting chain provides forwardness and infiniteness. The OTPs will be generated simultaneously on the client and the server side and will be compared. After successful validation of OTP's an additional layer of security i.e. a security question has to be answered to be fully authenticated into the system. The proposed scheme can resist an off-line guessing attack because it uses strong passwords produced from strong hash functions

7. Design

Fig. 2. Registration page

Fig. 3. Login page

Fig. 4. Website login

Fig. 5. Input variables page

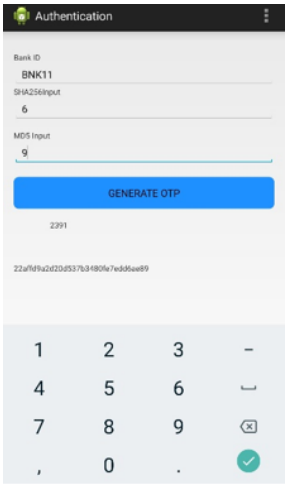

Fig. 6. Authentication page

Fig. 7. OTP validation page

8. Conclusion

A new multi-factor OTP-based authentication scheme has been proposed using mobile phones as they are becoming more and more powerful devices. It takes an advantage of increased processing power and the availability of the phone. Using two chain hashing functions we generate a OTP from an initial seed. We have illustrated our approach to an online authentication process for bank website. Our algorithm doesn't require a token embedded server. Our system overcomes the problems with utilizing OTPs with an SMS, consisting of the SMS cost and delay, along with international roaming restrictions like. Our algorithm doesn't require TOTP based system. The multi factor authentication property has been achieved without restrictions.

References

[1] L. Raddum, Nestås, K. Hole, "Security Analysis of Mobile Phones Used as OTP Generators", In: IFIP International Federation for Information Processing. 2010, pp. 324-331, 2015.
[2] N. Haller, "The S/KEY One–Time Password System. In: Proceedings of the ISOC Symposium on Network and Distributed System Security", pp. 151-157, 1994.
[3] K. Bicakci N. Baykal, "Infinite length hash chains and their applications" In: Proceedings of 1st IEEE Int. Workshops on Enabling Technologies: Infrastructure for Collaborating Enterprises WETICE'02, pp. 57-61, 2012.
[4] https://doubleoctopus.com/securitywiki/authentication/single-factor-authentication/
[5] https://en.wikipedia.org/wiki/Multi-factor_authentication
[6] Prathamesh S. Dhanorkar, Atish R. Jadhav "Multi-factor Authentication Using Mobile Phones" IJCSAI, Feb 2017.
[7] S.M. Siddique, M. Amir, "GSM Security Issues and Challenges Software Engineering," Artificial Intelligence, Networking and Parallel/Distributed Computing, 2016.