MULTISHARING DATA USING OTP

Ashwini Sukre[1], Saher Fakih[2], Shradha Shende[3], Rupali Tate[4], Sneha Bendale[5]

[1,2,3,4,5] *Department of Computer Engineering, Terna Engineering College,Mumbai University,India*

Abstract - *Online and Enterprise resources typically required authentication before user allow to access the sensitive applications and information. Sensitive information generally contains user personal information, transactions, confidential data, etc. Traditional user authentication system used user identifier, Password, Personal Identification Number (PIN), Token code etc. These Systems can't fulfill 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 multilevel and 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 paper, 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.*

Key Words: **Multisharing, OTP Generation, Authentication, Android, SHA256, MD5**

1. INTRODUCTION

Smartphones and tablets are quickly becoming the most ubiquitous general purpose computers in the world. Collaborative workers need to be able to share content and applications from their mobile devices on large shared displays. Need for securing the data over the internet is increasing. Authentication is the heart of every security model. Password based authentication is the most often utilized and trusted authentication mechanism. User authentication is often achieved by utilizing a single-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 major advantage of involving a mobile phone is that most users already have mobile phones, and therefore no extra hardware token needs to be bought, deployed, or supported. 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.

2. LITERATURE SURVEY

2.1 QR Code Based Secure OTP

OTP are used to provide higher layer of security over static passwords that are prone to replay attacks. Distribution of OTPs to concerned user is a major issue. Short message service that is available for mobile phones is the most common methodology for OTP distribution. Quick Response code (QR code) is actually two dimensional bar codes and can store information in both length and breath. QR codes are widely being used to convey short information such as website address, mobile numbers etc. In this paper we are presenting a new authentication scheme for secure OTP distribution in net banking through QR codes and email. [2]

2.2 Authentication Using TLS And Offline OTP

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 generate 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. [1]

2.3 Two Factor Authentication Using Smartphone

The two factor authentication implemented using SMS OTP or OTP generated by Smartphone- One Time Password to secure user accounts. The proposed method guarantees authenticating online banking features are secured also this method can be useful for e-shopping & ATM machines. The proposed system involves generating and delivering a One Time Password to mobile phone. Smartphone can be used as token for creating OTP or OTP can be send to mobile phone in form of SMS. The generated OTP is valid for only for short period of time and it is generated and verified using Secured Cryptographic Algorithm. [3]

2.4 Device Registration and Dynamic QR code based OTP Generation

Traditional user authentication system used user identifier, Password, Personal Identification Number (PIN), Token code etc. These Systems can't fulfill 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. 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. [6]

3. Existing System

Some works have proposed computation models for trust by incorporating the concept of risk. Like trust, reputation has also been studied extensively. No work addresses the issue of selecting trustworthy service provider in server marketplace. Estimation of risk of outsourcing a business onto third-party server has not been handled in reported works. Models proposed in reported works lack experimentation and analysis. In the Existing system can not provide any password therefore not secure.

3.1 Disadvantages of Existing system

- By sharing storage and networks with many other users/customers it is possible for other customers to access your data
- Single point of failure can occur during logging into websites.
- The authenticity is limited to known or managed laptops/desktops.
- We need to monitor application upgrades and changes.
- It has the lowest level of security.
- Unavailability of service due to low signal strength and coverage area.
- Unavailability of device as user needs to have the device physically present during the login process.

4. PROPOSED SYSTEM

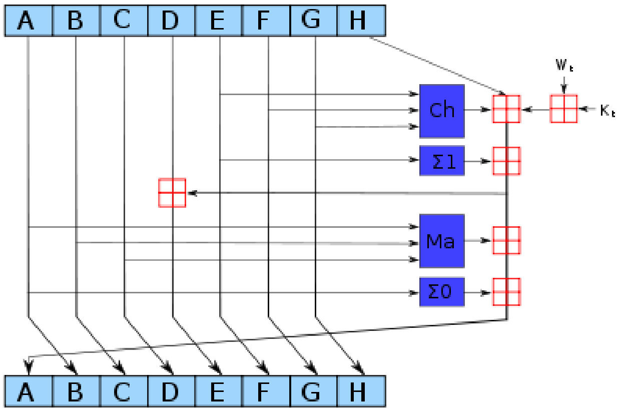
All User registers on the website via mobile app. The parameters during registration phase are username, password, security question Using Android's permission, the mobile app reads and sends the IMEI and IMSI number to the server. Server receives all the input data and creates an account for the users. Server will display the above input data on the mobile app which was used to create an account. Server will concatenate IMEI, IMSI to generate a seed. Users will now request login into website using username and password. 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 equal, then start the downloading the file.

4.1 Advantages of Proposed system

- The data is less prone to snooping and Man-In-The-Middle Attacks.
- The system makes use of nested hash chaining, overcomes the problem of hashing collision.
- The mobile app is fully independent of SMS channel and internet channel for OTP generation.
- The system can resist offline guessing because it uses strong passwords and hash functions.
- This scheme uses forward hashing techniques, which eliminates small challenge attack completely.

5. ALGORITHMS

5.1 SHA-256 Algorithm



(Figure 1.1)

Step 1: Append Padding Bits

Message is "padded" with a 1 and as many 0's as necessary to bring the message length to 512 bits fewer than an even multiple of 512.

Step 2: Append Length

64 bits are appended to the end of the padded message. These bits hold the binary format of 64 bits indicating the length of the original message.

Step 3: Prepare Processing Functions

Each step t ($0 \le t \le 63$): Word expansion for Wt

If t < 16

Wt = tth 32-bit word of Mj

If 16 ≤ t ≤ 63

S0 = (Wt-15 rightrotate 7) (Wt-15 rightrotate 18) (Wt-15 rightshift 3)

S1 = (Wt-2 rightrotate 17) (Wt-2 rightrotate 19) (Wt-2 rightshift 10)

Wt = Wt-16 + S0 + Wt-7 + S1

Each step t (0 ≤ t ≤ 63):S0 = (A rightrotate 2) (A rightrotate 13) (A rightrotate 22)

maj = (A ^ B) (A ^ B) (B ^ C)

t2 = S0 + maj

S1 = (E rightrotate 6) (E rightrotate 11) (E rightrotate 25)

ch = (E ^ F) ((E) ^ G)

t1 = H + S1 + ch + Kt + Wt

(A, B, C, D, E, F, G, H) = (t1 + t2, A, B, C, D + t1, E, F, G)

Step 4: Prepare Processing Constants

SHA1 requires 64 processing constant words defined as:

K(t) = 0x5A827999

(0 <= t <= 9)

K(t) = 0x6ED9EBA1 (20 <= t <= 29)

K(t) = 0x8F1BBCDC (40 <= t <= 49)

K(t) = 0xCA62C1D6 (60 <= t <= 63)

Step 5: Initialize Buffers

Finally, when all 64 steps have been processed, set

H0 = H0 + A

H1 = H1 + B

H2 = H2 + C

H3 = H3 + D

H4 = H4 + E

H5 = H5 + F

H6 = H6 + G

H7 = H7 + H

Step 6: Processing Message in 512-bit blocks (L blocks in total message)

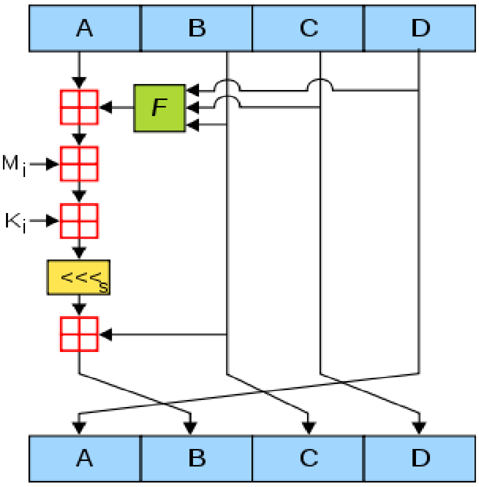
This is the main task of SHA1 algorithm which loops through the padded and appended message in 512-bit blocks. When all Mj have been processed, the 256-bit hash of M is available in H0, H1, H2, H3, H4, H5, H6, and H7.

Step 7: Pseudo Code

```
For loop on k = 1 to L
(W(0),W(1),...,W(15)) = M[k] /* Divide M[k] into 16 words */
For t = 16 to 79 do:
W(t) = (W(t-3) XOR W(t-8) XOR W(t-14) XOR W(t-16)) <<< 1
A = H0, B = H1, C = H2, D = H3, E = H4
For t = 0 to 63 do:
TEMP = A<<<5 + f(t;B,C,D) + E + W(t) + K(t) E = D, D = C,
C = B<<<30, B = A, A = TEMP
End of for loop
H0 = H0 + A, H1 = H1 + B, H2 = H2 + C, H3 = H3 + D, H4 = H4 + E
End of for loop
End of for loop
```

Step 8: Output H0, H1, H2, H3, H4, H5, H6, and H7: Word buffers with final message digest.

5.2 MD5 Algorithm



(Figure 1.2)

Step 1 : Append padded bits:

- The message is padded so that its length is congruent to 448, modulo 512.

- Means extended to just 64 bits shy of being of 512 bits long.

– A single “1” bit is appended to the message, and then “0” bits are appended so that the length in bits equals 448 modulo 512.

Step 2 : Append length:

– A 64 bit representation of b is appended to the result of the previous step.

– The resulting message has a length that is an exact multiple of 512 bits.

Step 3 : Initialize MD Buffer

- A four-word buffer (A,B,C,D) is used to compute the message digest.

– Here each of A,B,C,D, is a 32 bit register

- These registers are initialized to the following values in hexadecimal:

word A: 01 23 45 67

word B: 89 ab cd ef

word C: fe dc ba 98

word D: 76 54 32 10

Step 4 : Process message in 16-word blocks.

– Four auxiliary functions that take as input three 32-bit words and produce as output one 32-bit word.

F(X,Y,Z) = XY v not(X) Z

G(X,Y,Z) = XZ v Y not(Z)

H(X,Y,Z) = X xor Y xor Z

I(X,Y,Z) = Y xor (X v not(Z))

- if the bits of X, Y, and Z are independent and unbiased, the each bit of F(X,Y,Z), G(X,Y,Z), H(X,Y,Z), and I(X,Y,Z) will be independent and unbiased.

Step 5 : Output

– The message digest produced as output is A, B, C, D.

– That is, output begins with the low-order byte of A, and end with the high-order byte of D.

6. IMPLEMENTION

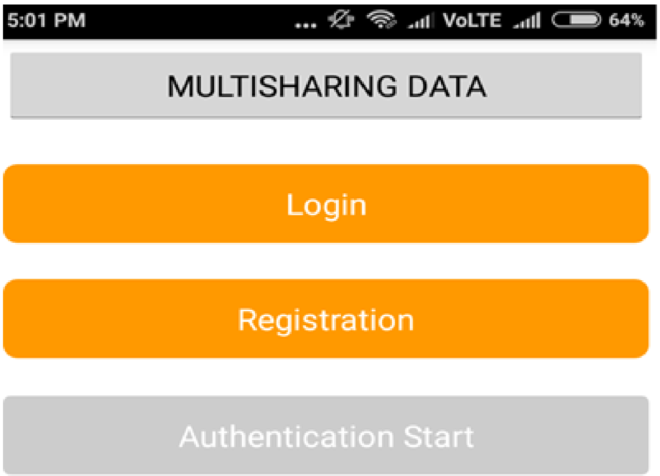
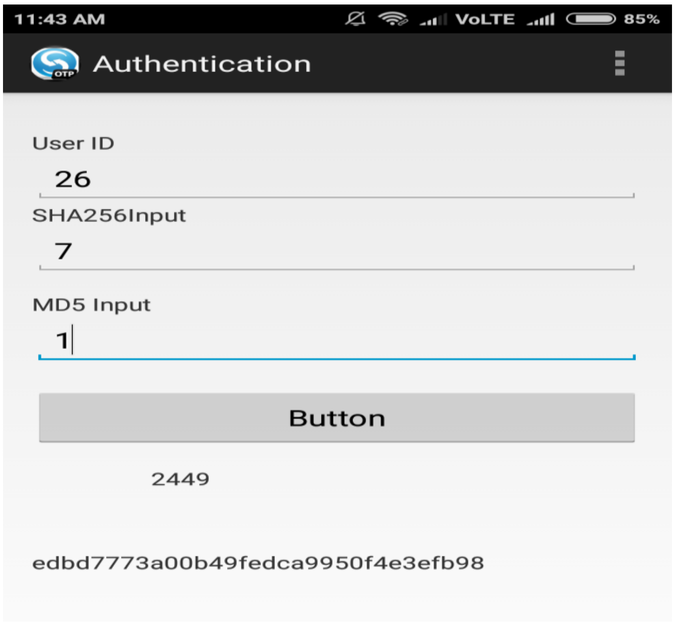


Figure 1.3: Registration And Login Page



User ID
26
SHA256Input
7
MD5 Input
1

2449

edbd7773a00b49fedca9950f4e3efb98

Figure 1.4: Authentication Page

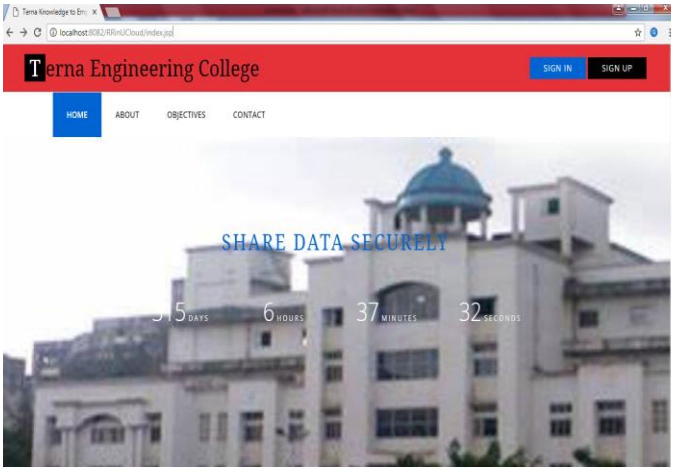
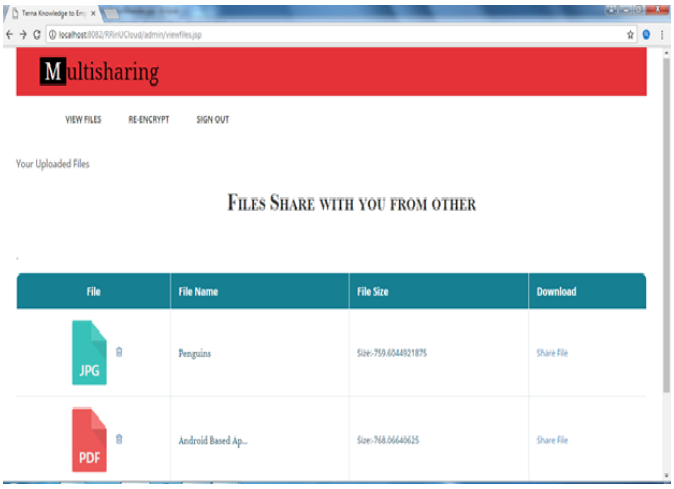


Figure 1.5: Home Page



Your Uploaded Files

Files Share with you from other

File	File Name	File Size	Download
JPG	Penguins	Size:-759.6044921875	Share File
PDF	Android Based Ap...	Size:-768.06640625	Share File

Figure 1.6: File Sharing Page

7. CONCLUSIONS

Data sharing opportunities must be carefully considered in the design of many-core chips targeting multi-threaded workloads. 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 the OTP for authenticating the user.

It never transmits the pre-shared number and the address of the device during the token generation process. server 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-sharing authentication security. In future, the factor required to identify and authenticate the device and user can be improved to enhance the security level.

REFERENCES

[1] Vishal Gangwar, Ravishankar, Dr Ashish kr. Luhach,” Mobile based Secure Authentication Using TLS and offline otp.” IJCTA, International Science Press pp.5253-5262

[2] Abhas Tandon, Rahul Sharma, Sankalp Sodhiya, P.M. Durai Raj Vincent,”QR code based Secure OTP distribution scheme for Authentication in Net-banking.” IJET, International Journal of Engineering and Technology.

[3] Sagar Acharya, Apoorva Polawar, P.Y.par,”Two Factor Authentication Using Smartphone.” IOSR Journal of Computer Engineering(IOSR-JCE), e-ISSN:2278-0661,p-ISSN:2278-8727

[4] 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.

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

[6] Deepak R. Thorat1 , Sheetal S. Sonawane2 “Risk Based Multilevel and Multifactor Authentication using Device Registration and Dynamic QR code based OTP Generation.” International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 10, October 2014

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.