A Framework for User Authentication and Authorization using Request based One Time Passkey and User Active Session Identification

Boopathy D.
Research Scholar
Department of Information Technology
Bharathiar University
Coimbatore – 641046, Tamilnadu.

M. Sundaresan, PhD
Professor and Head
Department of Information Technology
Bharathiar University
Coimbatore – 641046, Tamilnadu.

ABSTRACT

One-time password is currently used as one of the user authentication mechanisms. To avoid the username and password vulnerability, the two-way authentication mechanism has come into being, to provide security to the user at the login time. Many online service providers are using the two-way authentication mechanism as a key to identify whether the login user or service request person is a right one or not. To add more security to the user session, the Session Identification (SID) has been used. The user authentication and the user authorization are important for online transactions and web-related transaction services. Existing OTP methods are widely used by many service providers as it is, or with a little modification. This paper proposes Request-based One-Time Password (ROTP) as a new type of OTP mechanism and in the SID, the ROTP value is used as Active Session Identification (ASID) value. Inside Data Ownership Country Access (IDOCA) and Outside Data Ownership Country Access (ODOCA) data access permission rights are assigned to authorize the users. The proposed method satisfies the evaluation parameter and gives the satisfied result in the testing level environment.

Keywords

Session Identification, One-Time Password, User Authentication, Web Services, Cloud Security.

1. INTRODUCTION

The simple user authentication is based on the username and password verification. Once the verification has been done, then the user will be allowed to access the information from storage. When a storage system has not been connected to a network, the simple authentication mechanism is enough. When the system is connected to a network of computers, servers and storages, the simple user authentication mechanism will meet the system compromise. Nowadays most of the users have belief on the online storage as it is more efficient than local storage, because the user can access the information stored online from anywhere, any time.

For online storage data access service, the simple authentication mechanism is not enough. To create the user's trust and ensure the user's security the One-Time Password (OTP) [1] [2] methods are widely used by many online services. That service includes online purchase order confirmation, any kind of ticket booking confirmation, banking transaction confirmation etc. At the same time, once the user has been identified as a right person, then the service provider needs to create and assign the secured session to the verified user. In that allocated session only the users need to finish their requirement. To identify the user's session identification, the unique session identification value is used. Different programming languages use different value assigning methods for user session identification [3] [4]. Authentication is to prove that something is genuine and authorization is to give permission to someone. Once the username and password are verified then the Session Identification SID [5] [6] algorithm prepares one unique session identification number and assigns it to the user. So, the hacker cannot easily steal the information from the user due to the session. The user can get new session at each and every time of new login.

2. REVIEW OF LITERATURE

HOTP algorithm relies on two basic things; they are shared secret and a moving factor. HmacSHA1 hash of the moving factor will be generated using the shared secret. HOTP algorithm is event-based, meaning that whenever a new OTP is generated, the moving factor will be increased; therefore the subsequently generated passwords should be different each and every time [1].

TOTP algorithm works like HOTP. The TOTP algorithm also relies on a shared secret and a moving factor; however the moving factor works a bit different. In the case of TOTP, the moving factor constantly changes based on the time passed. The HmacSHA1 is calculated in the same way as with HOTP [7]. Google uses its OTP authentication for user validation. The OAUTH [8] (i.e. Open AUTHenticaion) is the standard used by Google for their purpose [8]. The ASP.NET session identifier is a randomly generated number encoded [9] into a 24-character string consisting of lowercase characters from a to z and numbers from 0 to 5 [10].

The higher set session.hash_bits_per_character gives the shorter session_id, it will become by using more bits per character. The possible values are 4, 5, or 6 [11]. When using sha-1 for hashing (by setting ini_set ('session.hash_function', 1) the following session string lengths are produced by the three sessions. hash_bits_per_character settings: 4 - 40 character string, 5 - 32 character string, 6 - 27 character string.

Session identifiers should be at least 128 bits long to prevent brute-force session guessing attacks [12]. While naming cookies some programming languages use different methods like JSESSIONID (Java EE), PHPSESSID (PHP), and ASPSESSIONID (Microsoft ASP) [19].

The methods of delivering the OTP values are Text Messaging, Mobile Phones, Proprietary Tokens, Web-Based Methods and Hard Copy. All the above-said OTP delivery methods have different issues. The OTP receiving methods create more cost of investment for OTP receiving device, it will create device dependency and if the device is lost, it will affect the user login too [13].

A session ID [15] [14] is a unique number that a Web server assigns a specific user for the duration of that user's visit i.e. session. The session ID can be stored as a cookie, as a form field, or as a Uniform Resource Locator. Some Web servers [16] [17] generate session IDs by simply incrementing static numbers. However, most servers use algorithms that involve more complex methods, such as factoring in the date and time of the visit along with other variables defined by the server administrator [3].

Session ID keys [13] [18], in their conventional form, do not offer secure Web [19] browsing. In existing methods skilled hackers can acquire session IDs by using the process called session prediction, and then masquerade as authorized users in a form of attack known as session hijacking [3]. In all of the existing programming language the unique session identification number will contain 16 to 32 digit values [25].

3. SECURED CLOUD DATA STORAGE PROTOTYPE MODEL (SCDSPM)

Figure 01 shows the Secured Cloud Data Storage Prototype Model. The Secured Cloud Data Storage Prototype Model’s sub-models and its connectivity’s are shown in figure 01.

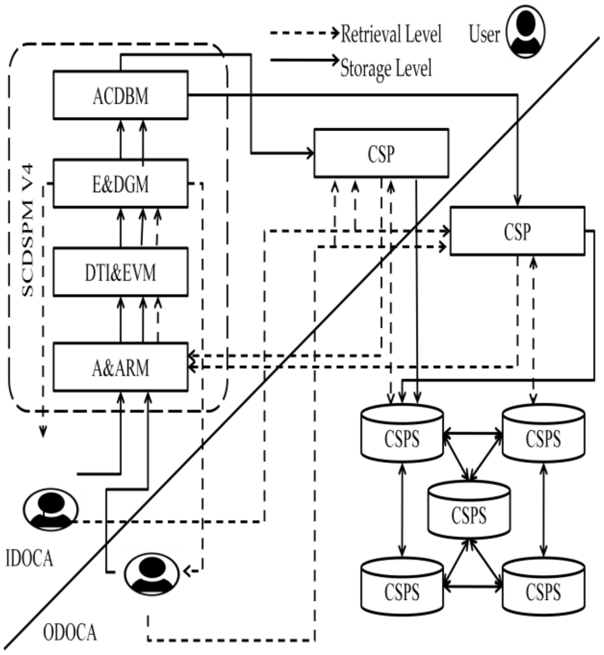


Fig 01: Secure Cloud Data Storage Prototype Model Version

The Secured Cloud Data Storage Prototype Model (SCDSPM) [20] [21] [22] contains four sub-modules; they are Authentication Authorization Resolving Module (AARM), Data Type Identification and Extension Validation Module (DTI&EVM), Encryption and Decryption Gateway Module (E&DGM) [23] [24] and Automatic Cloud Data Backup Module (ACDBM).

4. REQUEST BASED ONE TIME PASSWORD WITH ACTIVE SESSION IDENTIFICATION

This research paper deals with the SCDSPM’s first module. In the AARM the user identification and user authentication-related things are taken into consideration.

For that purpose the Request-based One-Time Password (ROTP) [25] and Active Session Identification (ASID) was designed and used to identify and authenticate the user in Authentication Authorization Resolving Model (AARM). The AARM needs to be coupled with the SCDSPM’s other sub-models to process the finite work. The AARM process includes,

- Username and Password Verification
- IP Address Location Identification
- Request-based One-Time Password Generation
- Request-based One-Time Password Validation
- IP Address and Mobile Number Cross Verification
- Outside Data Ownership Country Access (ODOCA) Data Handling Permission Assigning
- Inside Data Ownership Country Access (IDOCA) Data Handling Permission Assigning
- Active Session Identification Unique Value Assigning

The first sub-model of SCDSPM’s AARM output is forwarded to the next sub-model of SCDSPM Data Type Identification & Extension Validation Model (DTI&EVM) as input information. The figure 02 shows the AARM’s processes.

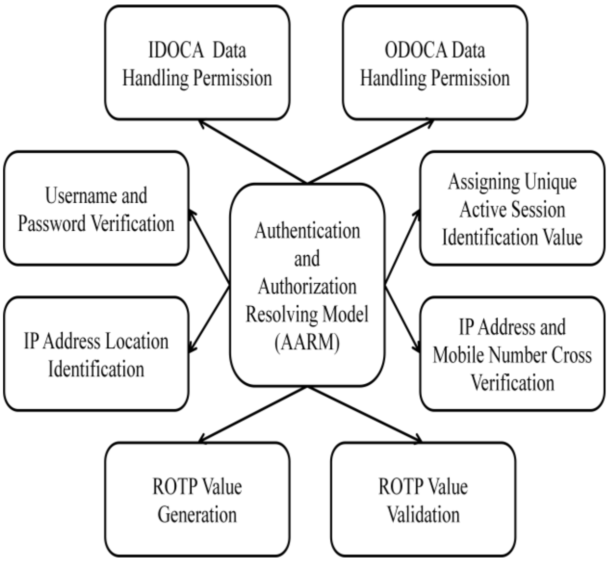

IDOCA Data Handling Permission
ODOCA Data Handling Permission
Username and Password Verification
Authentication and Authorization Resolving Model (AARM)
Assigning Unique Active Session Identification Value
IP Address Location Identification
IP Address and Mobile Number Cross Verification
ROTP Value Generation
ROTP Value Validation

Fig 02: Processes of Authentication and Authorization Resolving Model

4.1 Request based One Time Password

The Request-based One-Time Password (ROTP) uses alphabets, numbers and special characters to generate the ROTP value. Totally 75 characters are used in this ROTP value generation method. More number of characters will give more number of combinations of ROTP value and also more number of complex combinations.

$$\boxed{\textbf{ROTP value = Number of ROTP value in digits}^{\textbf{(Number of ROTP combination characters)}}}$$

For example if the ROTP value is 8 digits,

$$\boxed{\textbf{ROTP value} = \mathbf{8^{(75)}}}$$

So the total combination will be

$$\boxed{\begin{array}{c}\mathbf{8^{(75)} = 5.39198930E67} \\ \textbf{ROTP value complex combinations}\end{array}}$$

The user is permitted to enter the correct ROTP value with in 3 attempts or the user account will be suspended. So the total number of combinations will be divided by three to get the number of possibility to predict the approximate ROTP value.

$$\mathit{Average\ ROTP\ Value\ Prediction} = \frac{\mathit{Total\ ROTP\ Complex\ Value}}{\mathit{Tota\ number\ of\ attempts}}$$

So the total number of maximum possibility to predict the number for every login attempt will be

$$\frac{\mathbf{5.39198930E67}}{\mathbf{3}} = \mathbf{1.7973397E57}\ \textbf{Average ROTP value prediction}$$

Pseudo code for ROTP value Validation from user

```
for (i=0;i>=2;i++)
{
  if
     g(x) = f(x) assign 1 ——▶a
  else
     assign 0 ——▶a
}
assign -1 ——▶a
```

Generated ROTP value = g(x), User entered ROTP value = f(x), i = number of attempts by the user, a = process validated and it will be transferred to next process, 1 = transfer the process to next level, 0 = send error message to user, -1 = user access denied.

4.2 Active Session Identification

Once the user has entered the correct username and password, then the ROTP value will be sent to the user for authentication. When the user has entered the ROTP value for validation, the ROTP validation will be processed on the Server end. The existing unique session identification model is used in the different algorithms and different methods to generate and fix the value for the active session. But in this AARM model it has not used the separate algorithm to prepare the unique session identification number. Instead of using the separate algorithm, the ASID method uses the combination of ROTP value with the validated date and time of the ROTP value at server end as ASID value.

For example

Active Session Identification process

ROTP value is **#M#$Ib>_** and validation time of the ROTP value at server end will be **25-05-2016 16.35.46** then the ASID value will be **#M#$Ib>_25-05-2016 16.35.46**

4.3 Base64 Encode and Decode

Base64 is a group of similar binary-to-text encoding schemes that represent binary data in an ASCII format by translating it into a radix-64 representation. In the SCDSPM's AARM each and every data will be encoded before that information transfer from the user end to the server end. The username, password and user's IP address are also encoded before that information transfer to the server end. In the server end the encoded information will be decoded and then that decoded information is used for the validation. The same encoding concept is used in the ASID value fixing time also. At first the ROTP value and ROTP value validated time in server end will be encoded separately. After that encoding, both the encoded values will be merged and once again the merged value will be encoded. After that the merged encoded value will be fixed as ASID value for the user's active session.

ROTP value & Active Session Identification using Base 64

ROTP value = **#M#$Ib>_**

Encoded ROTP value = **I00jJEliKV8**

ROTP value validated timestamp = **25-25-2016 16:35:46**

Encoded ROTP value validated timestamp = **=MjUtMDUtMjAxNiAxNjozNToZNTo0NG==**

Merged Encoded ROTP value and ROTP value validated timestamp = **I00jJEliKV8=MjUtMDUtMjAxNiAxNjozNToZNTo0NG==**

Encoded value of Merged Encoded ROTP value and ROTP value validated timestamp =

=STAwaKPFbGlLVjg9TWpVdE1EVXRNakF4TmlBeE5qb3POVG8wTmc9PQ==

4.4 Data Handling Permission

Data handling permission is enabled only when the rights are given to the users. The user's data handling permission will be assigned to the users based on their IP address-related location information. If the user is trying to login into their account within their country boundary limit, then the data permission assigning process will assign Inside Data Ownership Country Access (IDOCA) to the user, else the data permission assigning process will assign Outside Data Ownership Country Access (ODOCA) to the user. If the IP address is unable to be traced then their login process will terminated.

Pseudo code to cross verification of IP address location and Mobile number information for assigning the data handling permission to the user

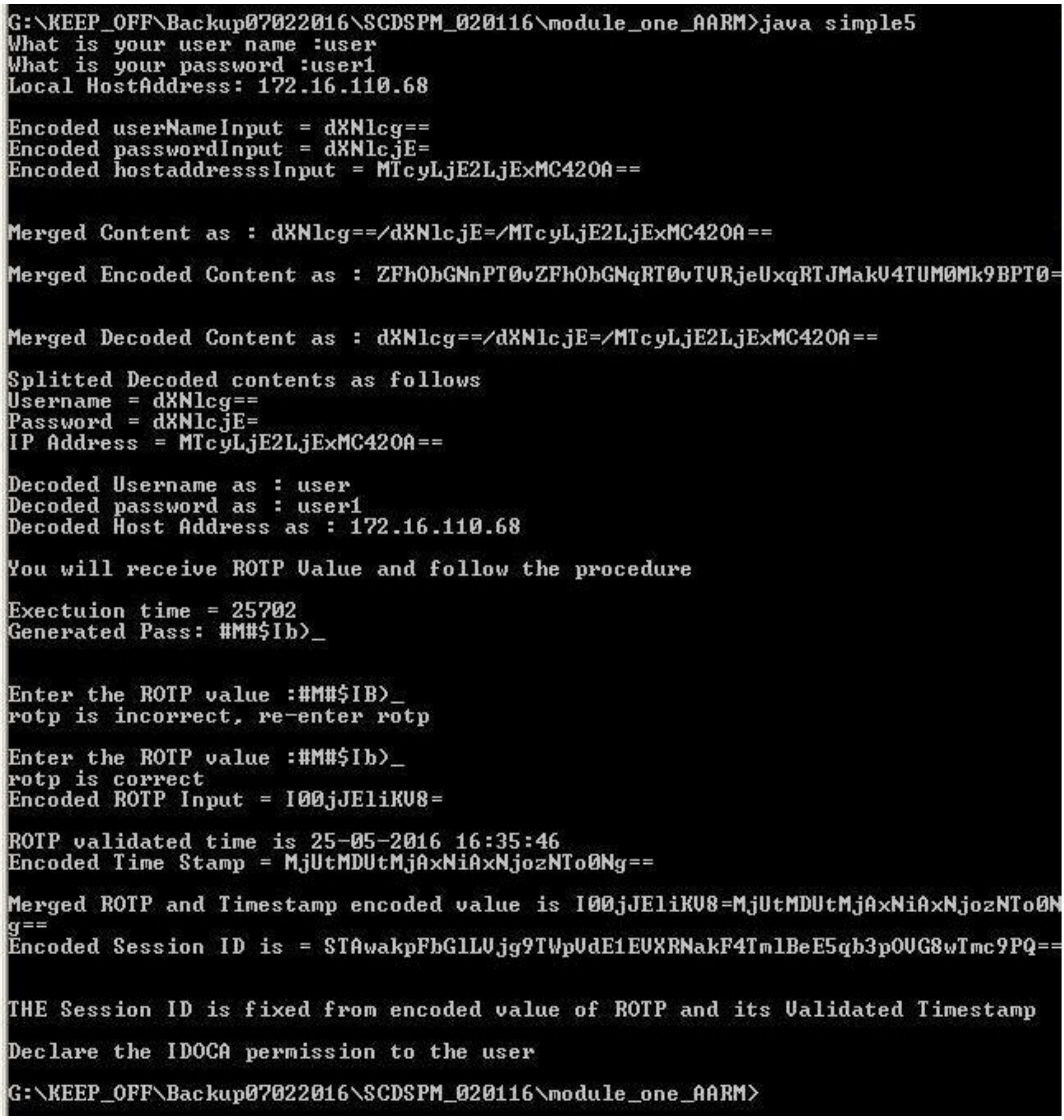
```
if f(x) = g(x) then
 {
   Assign Val 1 ——▶Data request user permission as
   IDOCA
 }
else
 {
   Assign Val 0 ——▶Data request user permission as
   ODOCA
 }
```

IP address location = g(x), Mobile Number location = f(x), Val 1 = IDOCA, Val 0 = ODOCA.

The above-explained Request-based One-Time Password, Active Session Identification, Base64 and Data Permission Assigning algorithms were used to process the user authentication, user validation using ROTP value, unique session identification value preparation and fixing it as user active session identification value and assigning the data handling permission to users.

5. RESULTS AND DISCSSION

The Java 1.8 SDK version was used to design the AARM's ROTP value Generation and Validation algorithm, Active Session Identification algorithm and Data handling permission assigning algorithm. Also the same java version was used to measure the time taken and resource utilization to process the above-mentioned AARM's algorithms. The figure 03 shows the sample output result for the AARM's algorithms processes.



```
G:\KEEP_OFF\Backup07022016\SCDSPM_020116\module_one_AARM>java simple5
What is your user name :user
What is your password :user1
Local HostAddress: 172.16.110.68

Encoded userNameInput = dXNlcg==
Encoded passwordInput = dXNlcjE=
Encoded hostaddresssInput = MTcyLjE2LjExMC42OA==


Merged Content as : dXNlcg==/dXNlcjE=/MTcyLjE2LjExMC42OA==

Merged Encoded Content as : ZFhObGNnPT0vZFhObGNqRT0vTVRjeUxqRTJMakV4TUM0Mk9BPT0=


Merged Decoded Content as : dXNlcg==/dXNlcjE=/MTcyLjE2LjExMC42OA==

Splitted Decoded contents as follows
Username = dXNlcg==
Password = dXNlcjE=
IP Address = MTcyLjE2LjExMC42OA==

Decoded Username as : user
Decoded password as : user1
Decoded Host Address as : 172.16.110.68

You will receive ROTP Value and follow the procedure

Exectuion time = 25702
Generated Pass: #M#$Ib)_


Enter the ROTP value :#M#$IB)_
rotp is incorrect, re-enter rotp

Enter the ROTP value :#M#$Ib)_
rotp is correct
Encoded ROTP Input = I00jJEliKV8=

ROTP validated time is 25-05-2016 16:35:46
Encoded Time Stamp = MjUtMDUtMjAxNiAxNjozNTo0Ng==

Merged ROTP and Timestamp encoded value is I00jJEliKV8=MjUtMDUtMjAxNiAxNjozNTo0N
g==
Encoded Session ID is = STAwakpFbGlLVjg9TWpVdE1EVXRNakF4TmlBeE5qb3pOVG8wTmc9PQ==


THE Session ID is fixed from encoded value of ROTP and its Validated Timestamp

Declare the IDOCA permission to the user

G:\KEEP_OFF\Backup07022016\SCDSPM_020116\module_one_AARM>
```

Fig 03: User Authentication using Request-based One-Time Password Method and User Active Session Identification Method

Table 1 shows the time taken to process all the AARM's algorithms (including ROTP value Generation and Validation algorithm, Active Session Identification algorithm and Data Handling permission assigning algorithm)

Table 2 shows the resource utilized to process all the AARM's algorithms (including ROTP value Generation and Validation algorithm, Active Session Identification algorithm and Data Handling permission assigning algorithm)

Table 1. Time taken to process the AARM's algorithms

S. No.	Processes Names	Time Taken to Process in Nanoseconds (Ns)
01	ROTP Generation	25702 Ns
02	Encode ROTP value entered by user	1559975 Ns
03	Decode ROTP value entered by user at server end	1495442 Ns
04	ROTP validation time-stamp generation	47525035 Ns
05	Encode validation time-stamp generation	98057 Ns
06	Encode ASID value	159518 Ns
Total Time taken to finish all the above Processes		**50863729 Ns**

Table 2. Resource Utilized to process the AARM's algorithms

S. No.	Processes Names	Total Resource Utilization in Bytes
01	ROTP Generation	**921872 Bytes**
02	Encode ROTP value entered by user	
03	Decode ROTP value entered by user at server end	
04	ROTP validation time-stamp generation	
05	Encode validation time-stamp generation	
06	Encode ASID value	

The total time taken to finish the AARM's processes was 50863729 Nanoseconds i.e. 50.863729 Milliseconds and the total resource utilized to finish the AARM's process was 921872 bytes i.e. 900.27 Kilobytes.

The existing methods are used by many cloud and online service providers to authenticate their users in different methods. But the AARM model was designed to provide the finite solution in single model to the authentication-related issues. This AARM algorithm provides the finite and complete process to users that include user verification, user authentication using ROTP value, Active Session Identification and data handling permission assigning. The AARM is one of the methods proposed to authenticate the users. The each authentication method is different from other authentication methods. The AARM is using complex methods to authenticate and authorize the users, so this AARM is not compared with other existing authentication methods. The processing time and resource utilization taken by the AARM to complete the authentication and authorization have been considered in this research paper.

6. CONCLUSION AND FUTURE ENHANCEMENT

AARM uses the ROTP value and ASID methods to verify the user credentials and get the user's IP address for authentication purpose. In authentication process, the IP address and user's mobile number are used to identify the location of the user by using cross verification. When the IP address is used for the identification purpose then automatically the session hijacking related issues will be avoided. The ROTP method is used to generate the complex ROTP value combinations. It will provide more number of complex ROTP value combinations and will automatically decrease the probability of predicting the random ROTP values. The ROTP value is verified at the SCDSPM's server end. The SCDSPM's server ROTP value verified time stamp is noted for further user. While using, AARM prepares the user's ASID value by using the complex ROTP value with the ROTP value validated server date and time. The ROTP value and ROTP validated server date and time were merged and fixed as a user ASID value. Before fixing the ASID value to the user session, ROTP value and ROTP value validated server time will be encoded using the Base64 encoding. These processes will increase more security and trust to the users. Also the hackers and data hijackers may need more time to predict the user's ASID value to reach the user's live sessions. Within that, the user transaction will come to an end.

This AARM is used to verify user's credentials, user authentication using ROTP value, Active Session Identification and data handling permission assigning. The output of the AARM will be forwarded to the next process for further use. Basically the AARM is one of the SCDSPM's sub models. The AARM needs to be coupled with the other sub-models of SCDSPM for complete process, so that only the SCDSPM's AARM will come to existence.

7. REFERENCES

[1] http://blogs.forgerock.org/petermajor/2014/02/one-time-passwords-hotp-and-totp/

[2] https://en.wikipedia.org/wiki/One-time_password

[3] Aaron Halfaker, Oliver Keyes, Daniel Kluver, Jacob Thebault-Spieker, Tien Nguyen, Kenneth Shores, Anuradha Uduwage, Morten Warncke-Wang, "User Session Identification Based on Strong Regularities in Inter-activity Time", International World Wide Web Conference Committee (IW3C2), WWW 2015, ACM 978-1-4503-3469-3/15/05, May 18–22, 2015, Florence, Italy pp- 410 – 418.

[4] http://searchsoftwarequality.techtarget.com/definition/session-ID

[5] C. E. Dinuca, D. Ciobanu, "Improving the Session Identification Using the Mean Time", International Journal Of Mathematical Models and Methods in Applied Sciences, Issue 2, Volume 6, 2012, Pp-265 – 272.

[6] Vijay Kumar Padala, Sayeed Yasin, Durga Bhavani Alanka, "A Novel Method for Data Cleaning and User-Session Identification for Web Mining" International Journal of Modern Engineering Research (IJMER), Vol. 3, Issue. 5, Sep - Oct. 2013, pp-2816-2819.

[7] http://blogs.forgerock.org/petermajor/page/2/

[8] http://blogs.forgerock.org/aggregator/category/oath/

[9] https://en.wikipedia.org/wiki/Base64

[10] http://stackoverflow.com/questions/861911/whats-the-size-of-an-asp-net-3-5-session-id

[11] http://stackoverflow.com/questions/12240922/what-is-the-length-of-a-php-session-id-string

[12] https://www.owasp.org/index.php/Insufficient_Session-ID_Length

[13] Young-Hwa An, "Security improvements of dynamic ID-based remote user authentication scheme with session key agreement," 2013 15th International Conference on Advanced Communications Technology (ICACT), PyeongChang, 2013, pp. 1072-1076.

[14] S. M. Kim, Y. H. Goo, M. S. Kim, S. G. Choi and M. J. Choi, "A method for service identification of SSL/TLS encrypted traffic with the relation of session ID and Server IP," 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), Busan, 2015, pp. 487-490.

[15] https://en.wikipedia.org/wiki/Session_ID

[16] Priyanka Patel, Mitixa Parmar, "Review on User Session Identification through Web Server Log", International Journal of Computer Science and Information Technologies, Vol. 5 (1), 2014, pp-146-148.

[17] Priyanka Patel, Mitixa Parmar, “Improve Heuristics for User Session Identification through Web Server Log in Web Usage Mining”, International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, pp-3562-3565.

[18] R. Tahir, H. Hu, D. Gu, K. McDonald-Maier and G. Howells, "A Scheme for the Generation of Strong ICMetrics Based Session Key Pairs for Secure Embedded System Applications," 2013 27th International Conference on Advanced Information Networking and Applications Workshops, Barcelona, 2013, pp. 689-696.

[19] X. Wang, M. J. Sheng, Y. Y. Lou, Y. Y. Shih and M. Chiang, "Internet of Things Session Management Over LTE—Balancing Signal Load, Power, and Delay," in IEEE Internet of Things Journal, vol. 3, no. 3, pp. 339-353, June 2016.

[20] Boopathy.D and Dr.M.Sundaresan, “Secured Cloud Data Storage – Prototype Trust Model for Public Cloud Storage”. Proceedings of International Conference on Information and Communication Technology for Sustainable Development – Volume I, Springer AISC, Volume 408, ISSN 2194-5357. Online ISBN 978-981-10-0129-1, ISBN 978-981-10-0127-7, DOI: 10.1007/978-981-10-0129-1_35, CSI Ahmadabad chapter and ACM Udaipur Chapter, The Pride Hotel, Ahmadabad, 03 – 04, Jul 2015, pp 329 - 337.

[21] Boopathy.D and Dr.M.Sundaresan, “Securing Public Data Storage in Cloud Environment”. ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India – Volume I, Springer AISC, Volume 248, ISBN 978-3-319-03107-1, Visakhapatnam, 13 – 15, Dec 2013, pp 555 -562.

[22] Boopathy.D and Dr.M.Sundaresan, “Policy Based Data Encryption Mechanism Framework Model for Data Storage in Public Cloud Service Deployment Model”. Proceedings of 2013 Elsevier Fourth International Joint Conference on Advances in Computer Science (AET 2013), ISBN 978-93-5107-193-8. Haryana, India, 13 – 14, Dec 2013, pp 423 – 429.

[23] Boopathy.D and Dr.M.Sundaresan, “Enhanced Encryption and Decryption Gateway Model for Cloud Data Security in Cloud Storage”. Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of Computer Society of India – Volume II, Springer AISC, Volume 338, ISSN 2194-5357, ISBN 978-3-319-13730-8, DOI: 10.1007/978-3-319-13731-5_45, Hyderabad, 12 – 14, Dec 2014, pp 415 - 421.

[24] Boopathy.D and Dr.M.Sundaresan, “Data Encryption Framework Model with Watermark Security for Data Storage in Public Cloud Model”. Proceedings of 2014 IEEE Eighth International Conference on Computing for Sustainable Global Development (INDIACom - 2014), ISSN 0973-7529 ISBN 978-93-80544-11-3, Bharati Vidyapeeth’s Institute of Computer Applications and Management (BVICAM), New Delhi, 05 – 07, Mar 2014, pp 1040 – 1044.

[25] Boopathy.D and Dr.M.Sundaresan, “Framework Model and Algorithm of Request based One Time Passkey (ROTP) Mechanism to Authenticate Cloud Users in Secured Way”. In: 3rd International Conference on “Computing for Sustainable Global Development (INDIACom-2016)”, ISSN 0973-7529; ISBN 978-93-80544-20-5, Bharati Vidyapeeth’s Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA), 16 – 18, Mar 2016, pp 5317 - 5322.