© 2018 IJSRSET | Volume 4 | Issue 4 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

# A System for Preventing Session Hijacking With a Token Based Fast Cookie Authentication

**Khevana Shah**
Information Technology, L.D. College of Engineering, Gujarat Technological University, Ahmedabad, Gujarat, India

## ABSTRACT

"The direct or indirect utilization of social knowledge or trust relationships in human-computer authentication systems deployed in online or offline contexts." A user authentication scheme that uses any form of social knowledge, utilizes users' trust relationships, monitors users' social contexts, or records users' friend associations for granting or denying access to any resource is considered a social authentication scheme. In this study we analyze the security in basic prospective of type of authentication, possible threats and terms to protect authentication from that threat. The use of insecure cookies as a means to authenticate web transactions in collaborative and social media websites presents a hazard to users' privacy. By proposed methodology we aim to provide higher level of security from this threats.

**Keywords:** Security, Session Hijacking, Cookie, Session cost, Number of transactions

## I. INTRODUCTION

### A. Session Hijacking

Many collaborative websites utilize session cookies as a cheaper alternative to the wide utilization of the secure HTTPS protocol. The unprotected nature of cookies can compromise the collaborative environment.

Evidently, the availability of this websites is extended to long durations has made this issue even more pressing.

The issue of session hijacking or 'side jacking' due to sniffing out of Internet cookies is one of the important Internet security concerns.

Session hijacking results from unlawful control over cookies during an ongoing internet session in an unprotected network where plaintext traffic is unencrypted.

Cookies are vulnerable to attacks, which makes their current deployment questionable and warrants a search for more reliable and secure techniques.

### B. Session Hijacking Attack

"The exploitation of the web session control mechanism, which is normally managed for a session token. Because http communication uses many different TCP connections, the web server needs a method to recognize every user's connections." The most useful method depends on a token that the Web Server sends to the client browser after a successful client authentication.

A session token is normally composed of a string of variable width and it could be used in different ways, like in the,

I) The URL,
II) The header of the http requisition as a cookie,
III) Other parts of the header of the http request,

IV) Body of the http requisition.

The rest of this paper proceeds as follows. In Section 2, we present our proposed scheme in detail. In Section 4, we discuss the implementation of our secure scheme and its performance. In Section 5, we review and examine existing cookie schemes. We give concluding remarks in Section 6.

## II. PPROPOSED METHODOLOGY

In the survey we found a case study as the threat of security due to session hijacking which as in Banking system when banks want to transact with the central one which is Reserve Bank of India (RBI) in India, than they will request for a credential, central bank will provide the credential data for the client bank which can be used in certain time period only, else it will be expired.

In this scenario, they use third party server to share the credential data, but intruders / attackers are intelligent enough to steal the data, but what they cannot is a time frame, for which they keep live till client bank initiate its transaction and in between the attacker will get his transaction executed successfully.

Some of the limitations are found out after literature survey - Multiple method support can be applied with different evaluation criteria, Client timestamp can be used to generate unique value generation.

Algorithms can be applied with timestamp on collaborative website or social media to evaluate more accurate results for session cost with security.

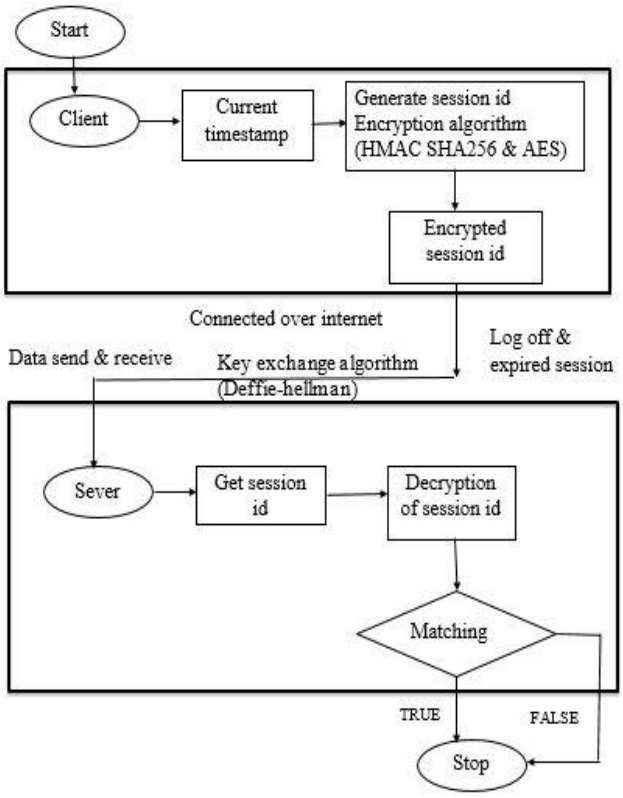


**Figure 1.** Flow diagram of the proposed system

**Algorithm (define the flow of the system)**

Input: Server connected over internet, to which the Client access information/ service using one's credentials.

I. Generation of Session ID using encryption algorithm
II. Encrypt the generated Session ID
III. Client send that encrypted Session ID to server through over the internet
IV. Decrypt the generated Session ID, using decryption algorithm
V. Get the Session ID using the same followed session executed
VI. Until the user log off or the session expired
VII. Exit.

We use Node.js technology for the implementation of this proposed system. The key feature of this technology are,

- ✓ Node.js is an open source server environment
- ✓ Node.js is free

- ✓ Node.js runs on various platforms (Windows, Linux, UNIX, Mac OS X, etc.)
- ✓ Node.js uses JavaScript on the server
- ✓ Moreover, it uses asynchronous programming.

## III. RESULTS AND DISCUSSION

Parameter Comparison to meet with the objectives to be analysed are,

**HTTP Session Persistence Frequency:**

The number of requests per minute received by the HADB depends on the persistence frequency. Persistence Frequency determines how often Application Server saves HTTP session data to the HADB.

The persistence frequency options are:

**1. web-method (default):** the server stores session data with every HTTP response. This option guarantees that stored session information will be up to date, but leads to high traffic to HADB.

**2. time-based:** the session is stored at the specified time interval. This option reduces the traffic to HADB, but does not guarantee that the session information will be up to date.

The following summarizes the advantages and disadvantages of persistence frequency options.

**1. web-method**

**Advantage:** Guarantees that the most up-to-date session information is available.

**Disadvantage:** Potentially increased response time and reduced throughput.

**2. time-based**

**Advantage:** Better response time and potentially better throughput.

**Disadvantage:** Less guarantee that the most updated session information is available after the failure of an application server instance.

1. **Average transaction cost * Number of transaction**

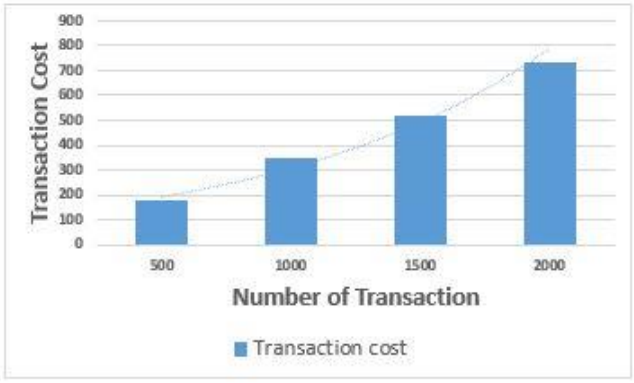


**Figure 2.** Parameter 1 as values in proposed system based on frequency of occurrence

**Table 1.** comparison with other method

| Number of transaction | Average transaction cost | |
|---|---|---|
| | Proposed method | Base paper method [1] Italic |
| 500 | 175 | 250 |
| 1000 | 350 | 500 |
| 1500 | 515 | 750 |
| 2000 | 735 | 1000 |

As we can analyse from the above mentioned chart and table that we produce a system with more security layer though it did not increase the transaction cost, other than that it help to gain the lesser value for the same.

2. **Average server response time for server side comparison**

Transpose = $(n / r) - T_{think}$

Where,

n = Number of concurrent user

r = Number of request per second

$T_{think}$ = Average think time (in seconds)

I.e. Response time = (5000 / 1000) – 3

= 5 -3

= 2 seconds

Server response time becomes steady at once, because previous requests are already processed.

## IV. CONCLUSION

Prevention against session hijacking is highly required to plays core part of the digitization in any application areas which involves the server-client communication. Now it is become important to design not only the secure method from the hijacking but to abstract the essence of that which are profitable and efficient for the future work. The proposed solution provide the significant Efficient and Accurate solution for the same without increasing the average server processing Time.

## V. REFERENCES

[1]. "Robust and Fast Authentication of Session Cookies in Collaborative and Social Media Using Position-Indexed Hashing" by Amerah Alabrah, Mostafa Bassiouni Year 2013, IEEE.

[2]. "Preventing session hijacking in collaborative applications with hybrid cache-supported one-way hash chains" by Amerah Alabrah and Mostafa Bassiouni, year 2014, in the IEEE.

[3]. Alex X. Liu, Jason M. Kovacs , Mohamed G. Gouda "A secure cookie scheme" elsevier, 2012 Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824-1266, USA Exis Web Solutions Department of Computer Sciences, The University of Texas at Austin, Austin, TX 78712-0233, USA DOI: 10.1016/j.comment.2012.01.013

[4]. "Handling TCP-Session Hijacking With Transport Layer Defence Method (TLD) In Mobile Adhoc Networks " by K.Geetha, Department of Computer Science, Periyar College, Cuddalore, India in ARPN Journal of Engineering and Applied Science, Vol.11, No.11,2016 ISSN 1819-6608.

[5]. "Social Authentication Applications, Attacks, Defence Strategies and Future Research Direction: A systematic review" by Noura Alomar, Mansour Alsaleh, Abdulrahman Alarifi, Year 2017, IEEE.

[6]. "A Survey on Detection Tools and Prevention Techniques for Session Hijacking Attack" by D.Madhavi, Assistant Professor, V.R.Sidhhartha Engineering College, Vijayawada, A.P., India, in International Journal for Scientific Research & Development Vol.2, Issue 12], 2015.

[7]. Joon S. Park and Ravi Sandhu, George Mason University "Secure cookie on the web" IEEE, 2002

[8]. Chuan Yue, Mengjun Xie, Haining Wang "An automatic HTTP cookie management system", ELSEVIER, 2010 Department of Computer Science, The College of William and Mary, Williamsburg, VA23187, United States DOI: 10.1016/j.comment.2010.03.006

[9]. Paul Rabinovich "Secure cross-domain cookies for HTTP", Springer, 2016 Security Software Development, Exostar, Herndon, USA.

[10]. J. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung, "Fourth factor authentication: somebody you know," in Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006, pp. 168–178.