CHAPTER 3

Preventing Session Hijack attacks in Web Applications using MAC appended Session ID

3. 1 INTRODUCTION

Current web applications are weakly secured against internet vulnerabilities such as broken key authentication, Cross Site Scripting attacks, session hijacking, buffer overflow attacks and Cross Site Request Forgery attacks. There are many existing approaches and methods are available to prevent the session hijack attack in web applications.

Due to the increased number of web applications over the World Wide Web, it is necessary for the web applications to be secured against web session hijack attacks. The key features such as security architecture, security flaws and integrity of the communication messages between client and web application server are taken into consideration.

3.2 PROPOSED SYSTEM ENVIRONMENT

The network that has a set of systems connected to the internet is configured. When a particular client in a network wants to establish the connection, it has to send a request to the web server to initiate the web session. The conditions of the proposed system environment are as follows:

(i) Each client is connected to network.
(ii) Each client has its own processors.
(iii) Each client has its own browsers.
(iv) When a client is connected to the web application the session states, session ID's and session attributes are monitored and recorded for decision making.

3.3 CONNECTION ESTABLISHMENT

User of the client has to establish the internet connection with the web server to access the web application as shown in the Fig.3.1

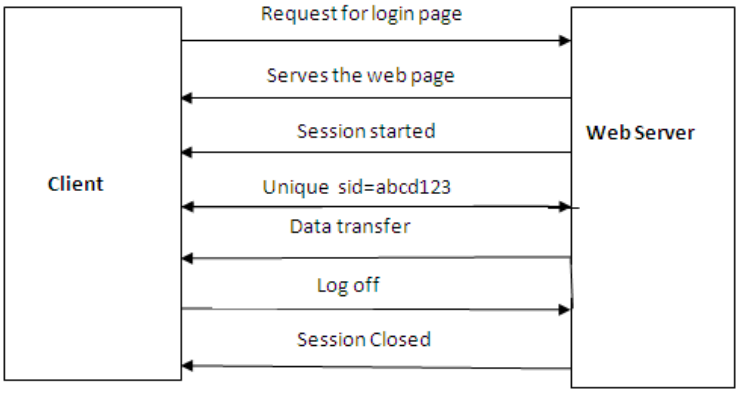

Fig.3.1 Connection Establishment

When the client requests the server for a session, the unique session ID is assigned to the web application session to transfer the data.

3.4 PROPOSED ARCHITECTURE

Session ID of three different lengths such as 160, 192 and 248 characters are generated and compared the efficiency and integrity of all the three methods. The three methods are the 160, 192 and 248 character length of session ID. In order to protect the integrity of the session ID's the Message Authentication Code (MAC) is used.

The architecture which is shown in the Fig.3.2 uses strong and MAC appended session ID to protect the integrity of the session ID during the web application session. First, a client establishes the connection with the server.

Brute force attack will crack the Message Authentication Code (MAC) if the length of MAC is less than 128 characters. So minimum of 128 character MAC or greater than 128

characters MAC is required to overcome from the brute force attack. Old session ID of 3 different lengths such as 32, 64 and 120 characters are chosen on ramdom basis.

SHA 512 hash algorithm is used to generate the Message Authtication Code (MAC) of 128 characters. The generated MAC of 128 characters is appended to old session ID of 32, 64 and 120 characters. The Session ID appeded with MAC will become the current Session ID of length 160 (32+128) characters, 192 (64+128), 248 (120+128) characters for the web application session. The client receieves the session ID from the server and uses the session ID until the session expires or closes. Message Authentication Code is used to verify the integrity of the Session ID.

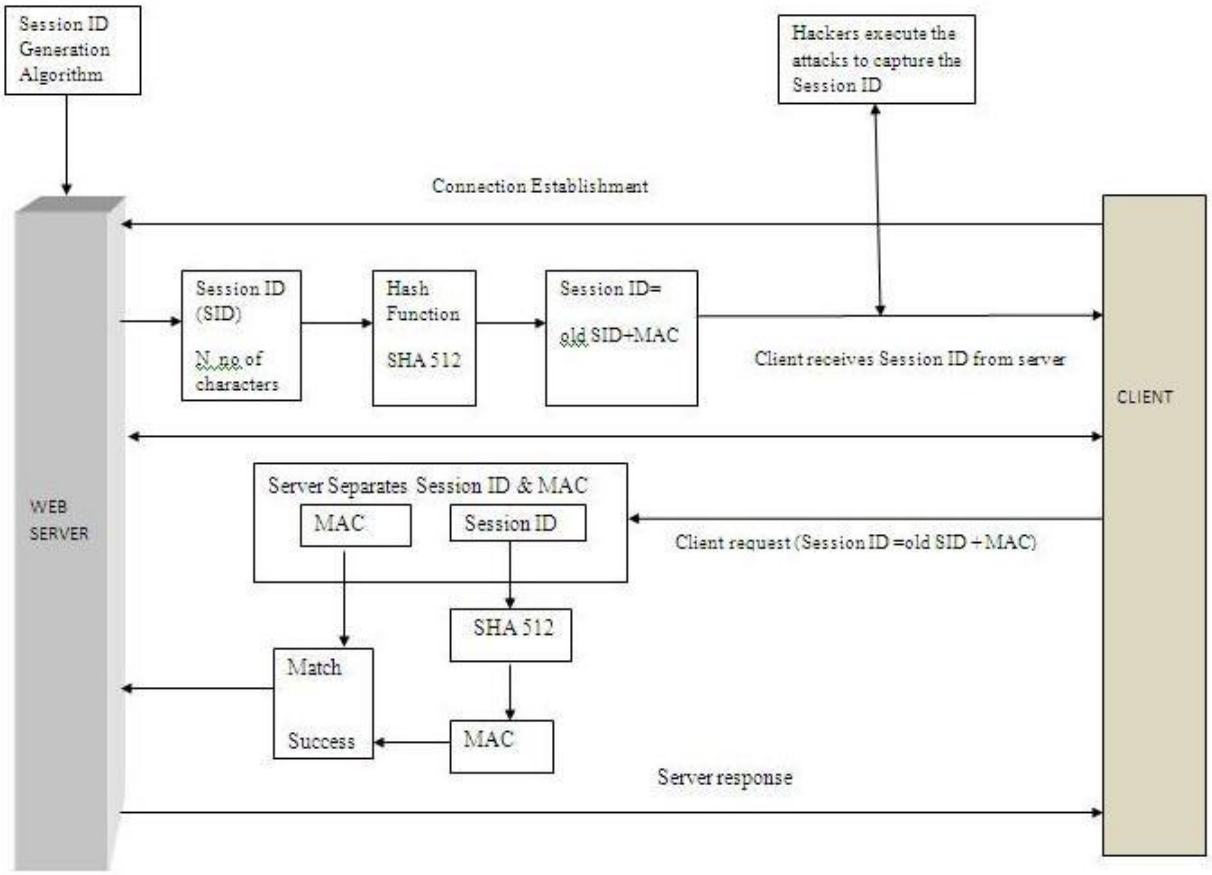

Fig.3.2 Proposed System Architecture

3.5 PROPOSED METHODS

Strong Session ID is used to strengthen the session and MAC to protect the integrity of the session ID in the web sessions. MAC appended Strong Session ID in three different cases are generated. The experiment was conducted for the following three cases.

Case 1 : The total length of the Session ID (SID + MAC) is 160 characters. i.e, old Session ID (SID) of 32 characters and Message Authentication Code (MAC) of 128 characters.

Case 2 : The total length of the Session ID (SID + MAC) is 192 characters i.e, old Session ID (SID) of 64 characters and Message Authentication Code (MAC) of 128 characters.

Case 3 : The total length of the Session ID (SID + MAC) is 248 characers. i.e, old Session ID (SID) of 120 characters and Message Authentication Code (MAC) of 128 characters.

For the above three cases, the MAC appended Session ID is generated. When the client is having the session with the web server, Network Sniffing, brute force attack and Cross Site Scripting attacks are executed to capture the Session ID. The experiments are focused on the attack results.

3.6 MAC APPENDED SESSION ID GENERATION

The following algortithm is used to generate the MAC appended Session ID. Java language can be used in Apache Tomcat server platform that generates the Session ID of required length and applies the SHA-512 Hash algorithm on the generated Session ID to obtain the Message Authentication Code (MAC). Session ID that is appended with MAC will become the current Session ID for the session.

Variables:

SID = randomly generated session id of length 32 or 64 or 120

MSID =modified session id after inserting special characters

HSID= SHA-512 hash value of MSID

FSID = final session id generated by our algorithm

CharList =list of special character [‘{‘,’&’,’*’,’%’,’^’,’#’,’@’,’!’]

Pseudo Code

```
Function generateSessionID ()
{
  SID← generate random session id of length 32 or 64 or 120
    If (SID==32 char) Then
       MSID← InsertSpecialChar(SID,3)
    Else if (SID==64 char) Then
        MSID← InsertSpecialChar(SID,4)
    Else if (SID==120 char) Then
         X←select random between 5 to 7
        MSID← InsertSpecialChar(SID,X)
   End If
 HSID=SHA-512(TempSID)
 FSID=MSID+HSID
 }

Function InsertSpecialChar (SID, K)
{
        Len← length of SID
        For counter  1 to K Do
I ←generate random number between 0 to Len
C ←get random special character from CharList
SID← Insert C at index I in SID
        End Do
}
```

3.6.1 MAC APPENDED SESSION ID GENERATION FOR N CHARACTERS

Session ID of 32, 64 and 120 characters are generated. SHA-512 hash algorithm is applied on the 32 ,64 and 120 characters Session ID to get the MAC appended session ID.

Algorithm :

(1) Client login( )

(2) Generate Session ID(n)

where n = 32, 64 and 160 characters

(3) MAC (128) ← SHA 512{Session ID}

(4) Session $ID_{current}$= Session ID + MAC

(5) Client ← Session $ID_{current}$ (n); n =160, 192 and 248 characters

(6) While client receives Session $ID_{current}$

(7) Capture Session ID ( )

(8) execute packet sniffing ( )

(9) execute brute force attack ( )

(10) execute cross site scripting attack( )

(11) $Server_{receieves}$ ← Session $ID_{current}$

(12) $Server_{separates}$: Session $ID_{separated}$

(13) $Server_{separates}$: MAC

(14) $MAC_{computed}$= SHA 512{Session $ID_{separated}$}

(15) If(match{$MAC_{computed}$, $MAC_{separated}$}=yes

(16) then

(17) server response the client request

(18) else

(19) server blocks the client request

3.7 EXECUTION OF ATTACKS

The following attacks are executed to capture the session ID. During the data trandsfer the attacks are executed and the results are recorded. No of session IDs captured, no of session IDs are not captured are observed for 10 sessions, 20 sessions and 30 sessions.

3.7.1 BRUTE FORCE ATTACK

Automated method of identifying the user's session ID by trial and error method.

```
Brute force attack
for each k=32 to k=200
Brute Force (K);
 end for
 Function Brute Force(int SIDLEN)
 {
       while (TRY ALL POSSIBILITIES)
      {
         SID = Generate random SIDLEN char session
         if (Try SID success)
            {
             break;
            }
           else
            {
             SID=new session of length SIDLEN;
             continue;
            }

       }
 }
```

3.7.2 CROSS SITE SCRIPTING ATTACK

```
Function Cross Site Scripting attack ( )
{
      while( Check URLs with hidden java scripts )
```

```
    {
       Generate random java scripts ( )

       if (URL with Java Script = sucess)
         {
          Set SID=URL_javascript
         }
         else
         {
          break;

         }

     }
}
```

3.7.3 MAN-IN-THE-MIDDLE ATTACK

```
Function Man in Middle attack ( )
{
     while(client login the session )
    {
         Execute Packet Sniffer( )
         Sniff the packets( )
         Capture Session ID ( )
      }
}
```

3.8 SEPARATING THE MAC FROM SESSION ID

The following pseudo code is used to separate the MAC that is appended with generated Session ID.

Variables

FSID = final session id generated in previous algorithm

Len =length of final session id

HLen= length of hash value (128 char)

OSID=session id after separating MAC (hash)

MAC= Hash value separated from FSID

Pseudo Code

```
Function Separate (FSID, HLen)
{
    Initialize OSID←NULL
    Len← Length of FSID
    X←Len – Hlen
     For counter 0 to X – 1 Do
          OSID=OSID + Read each char and append
      End do
      For counter X to Len Do
            MAC=MAC + read each char and append
       End do
}
```

3.9 IMPLEMENTATION ENVIRONMENT

Apache tomcat server is used to create the web application. Java language is used as source language to create the web application. A sample web application www.royalbank.com is developed with Tomcat server. In the developed web application, the web client is authenticated by the server using the client's login credentials. The web client establishes the session with the web server and server assigns the session ID to the client. Packet sniffing, brute force attack and Cross Site Scripting attacks are executed to capture the session ID during the transfer of Session ID to the server by the client.

3.10 SECURITY METRICS

To measure the performance of the proposed system the following security metrics are measured for the 3 cases of the generated session ID.

3.10.1 NUMBER OF SPECIAL CHARACTERS IN THE SESSION ID

Each time a client logs into the session, the server assigns the unique session ID for the client. The number of special characters in the session ID is measured.

3.10.2 NUMBER OF CAPTURED OR ATTACKED SESSION IDs

During the transmission of the session ID, the hackers execute the attacks to capture the session ID. If the session ID is captured, then the session can be hijacked from the client.

3.10.3 NUMBER OF PREVENTED SESSION IDs

Even though hackers are executing the attacks to capture the session ID, session IDs of some sessions are not captured. This is called prevented session ID.

3.10.4 SESSION ID GENERATION TIME

The generation time of the session ID is measured.

3.10.5 SESSION HIJACK PREVENTION RATE

It is the ratio between the number of session IDs not captured and the number of unique session IDs generated for “n” number of sessions.

Number of session IDs generated for “n” sessions = a

Number of session IDs captured = b

Number of session IDs prevented = c

Session Hijack Prevention rate = P_r

$$P_r = c / a$$

3.11 RESULTS AND DISCUSSION

For experimental purpose, proposed system is tested with a network of 50 systems with a web server. The web application is hosted in the web server. Out of 50 systems, 3 systems acts as intruder systems in which each intruder system executes Brute force attack, Cross Site Scripting attack and Man-in-the-Middle attack respectively to capture the session ID.

We have tested the integrity of the session ID of 3 different lengths of MAC appended session ID such as 160, 192 and 248 characters by creating 10 web sessions, 20 web sessions and 30 web sessions between the web server and the clients. Number of session IDs captured and the number of session IDs are not captured are recorded for 10 sessions, 20 sessions and 30 sessions. The session ID is generated for the above 3 different cases. The results are discussed below.

3.11.1 160 CHARACTER MAC APPENDED SESSION ID

The server generates the 32 characters session ID and the Message Authentication Code (MAC) of 128 characters is appended to the session ID. The length of the current session ID is 160 characters. The attacks are executed to capture the Session ID. The integrity of the session ID is tested using MAC by creating the 10 web sessions, 20 web sessions and 30 web sessions between the client and the server. Examples of the 32 characters old Session IDs are given in Table.3.1

Table.3.1 Old Session ID of 32 characters

No	Old Session ID
1	5F4*C6C233025F42E80A6%FD5591225#498
2	4FF43$9318CE06CC#E45%A0D4209DC82A1B
3	88@F82297D5518E731%4DC3D4EC575AD$0F
4	73B671F7&55DAE789A#4559*B11AB5871EC

The generated Session ID is appended with MAC. The Table.3.2 shows MAC appended new Session ID.

Table.3.2 MAC appended Session ID of 160 characters

No	Current Session ID (old ID + MAC Appended) 160 characters
1	5F4*C6C233025F42E80A6%FD5591225#498f565df02de463e7c3d78c516e4fc103dbd9111269cc1486c069441102e760904e33ddc751ed33a5d218c4cd164567809e0a88f2e27fdee7659c9184e129f4b62
2	4FF43$9318CE06CC#E45%A0D4209DC82A1B30cf14eba6fd8ca13d7d1cc16f821adef6d12d1f5bfd655c64f94767eaff28a007f8ad6bdfbcdeaa684ba1d93a9cf654f32d3c74e2cfdd820776d921b581bed7
3	88@F82297D5518E731%4DC3D4EC575AD0F8bff0b2a3967122d40e239ccbb1a5c0ecb05d19b2c6757404d2cb0ca17e69e3e7a6370b98df71f702be09c773ab5b90ff03bd1869fcb4a72dcdee2aced8e794
4	73B671F7&55DAE789A#4559*B11AB5871EC69ce8b3ed047b2acc720f80a234addbdbd758a67eecf97899e8c01c2e2e94a956efce735947526501c2bf2ffa3bfd7bef293a4971e8354e11d39d411b622de

The results of number of Session IDs attacked and the number of session IDs prevented are given in Table.3.3

Table.3.3 No of Session IDs captured for 160 characters

No	Attack Executed	No of Session IDs captured		
		10 sessions	20 sessions	30 sessions
1	Number of unique session IDs generated	10	20	30
2	Number of session IDs captured by attacks	0	1	1
3	Number of session IDs prevented from attacks	10	19	29
4	Session Hijack Prevention Rate	100 %	95 %	97 %

3.11.2 192 CHARACTER MAC APPENDED SESSION ID

The server generates the 64 characters old session ID and the Message Authentication Code (MAC) of 128 characters are appended to the session ID. The length of the current session ID is 192 characters. The attacks are executed to capture the Session ID. The integrity of the session ID is tested using MAC by creating the 10 web sessions, 20 web sessions and 30 web sessions between the clients with the server. Examples of the old session IDs are given in Table.3.4 and MAC appended Session ID of 192 characters are shown in Table.3.5

Table.3.4 Old session ID of 64 characters

No	Old session ID
1	0089E66*&E6153EF9D911245931BA%B64F9965DF1A7FBC8B05#A9DB60B96424FECD5
2	61E8&A780AAE08ABB7836B303*7660A8F9D4C296FDCF1D#9C0C97$492641F96286E6
3	7B63*C8F%95D14EE1F99C6402D692#DBC5DBD856B638E@EBBB&C0BE41D37C880841C6
4	59D632*D3A$B&2E34B79C903521BC@8D998753077D79940084%6389390D1F02781DFB

Table 3.5 MAC appended Session ID of 192 characters

No	Current Session ID (64 +128 MAC) = 192
1	0089E66*&E6153EF9D911245931BAB64F9965DF1A7FBC8B05#A9DB60B96424FECD5d0d8bd5e90aae304b2e06611ee4560f528c52e5334818883227f7d3ab80a689fdeb5c927e91527b8e60a9c936764fa27af0045b349842b0364fd212ce5f2b7fc
2	61E8&A780AAE08ABB7836B303*7660A8F9D4C296FDCF1D#9C0C97$492641F96286E6ab3a59f8ff072119990b6afbba9fc500a63bc553735736ed5a68e63eb8fdab417a26d8469c528c7bd26226a9f7d0d8e2148246b2b0595e955e50407a7e98377a

3	7B63*C8F95D14EE1F99C6402D692#DBC5DBD856B638E@EBBB&C0BE41D37C880841C617e6211bf13eaefb50c12f14f8813730a47d9d8ad120ebdf2ecdb63c4eb44a38ebf5fdd3e5b3be03b0ab03ae3c4828c47c689314c179767b398a8fce7b3f4f6f
4	$B&2E34B79C903521BC@8D998753077D79940084%6389390D1F02781DFBd47a47eed9bc14bbb6e3dfbbca00afdc647e8209ab3d4c4b13be84da7e457fbcf2f40341a5e1c177367fd551b8e20ec2b7030f61c556813316335854d3dfc48

Number of session IDs captured by attacks and the number of session IDs prevented by the attacks for 10 sessions, 20 sessions and 30 session are recorded in Table.3.6

Table.3.6 Number of Session IDs captured for 192 character MAC Session ID

No	Attack Executed	No of Session IDs captured		
		10 sessions	20 sessions	30 sessions
1	Number of unique session IDs generated	10	20	30
2	Number of session IDs captured by attacks	0	0	1
3	Number of session IDs prevented from attacks	10	20	29
4	Session Hijack Prevention Rate	100 %	100 %	97 %

3.11.3 248 CHARACTER MAC APPENDED SESSION ID

The server generates the 120 characters session ID and the Message Authentication Code (MAC) of 128 characters is appended to the session ID. The length of the current session ID is 248 characters. The attacks are executed to capture the Session ID. The integrity of the session ID is tested using MAC by creating the 10 web sessions, 20 web sessions and

30 web sessions between the clients with the server. The observed results for each session are given in Table.3.7 and the MAC appended Session ID of 248 characters are shown in Table.3.8

Table.3.7 Old Session ID of 120 characters

No	Old Session ID
1	6FF17@3E2CB22BBE#D203BD82D767D8C9E9E16AA5%70F00C22D6E19ED@C#3AE666934D9B57342D4CF865%8E211FA*2ADAA60EDA9FAE2F43323EAC123512EF14
2	FF3$0F12860F4F3EF7F4C9AEF815D4DA42$D6DCDD7359C6492B97077B0A1#DB93962@EB22DB399DBDB0DD80B551E9F*E50E38F6817A0DFFF#90910AD93F76F
3	F43C8D9#2507420EFA6B59B484FE961BA3DD688%36F01F8C426278C17C2@15B3A4152F3D0$6860C1B5890897$7A08288@C47DB0F0139B24A76C02AD8FAF168
4	D4ED2DC*EC653AC04EEC99025F802E36E16276665#385C4880434CE33A6@EC95A1ED3C848253B278881%A72$85DA24D636E0FBF1F4E6F826FC956FCC37D68

Table.3.8 MAC appended Session ID of 248 characters

No	Current Session ID (120+128) = 248 characters
1	6FF17@3E2CB22BBE#D203BD82D767D8C9E9E16AA5%70F00C22D6E19ED@C#3AE666934D9B57342D4CF865%8E211FA*2ADAA60EDA9FAE2F43323EAC123512EF149dec6c606d7c15864d79d6a5394aec9c7d9c748343c69bed270d6972c19e737c54d7d94d69686e639d2a4102202a87475f142e747fa53a24307c9088c8ef18f
2	FF3$0F12860F4F3EF7F4C9AEF815D4DA42$D6DCDD7359C6492B97077B0A1#DB93962@EB22DB399DBDB0DD80B551E9F*E50E38F6817A0DFFF#90910AD93F76Fd0628d354d06006a5750547cc82c3ab78502c7db8d2e3c7004de1be104b1419bb2cc9a603ce9c174ee6ad3f91bc15c063120740a479052fc608ec2825f4afb6

3	F43C8D9#2507420EFA6B59B484FE961BA3DD688%36F01F8C426278C17C2@15B3A4152F3D0$6860C1B5890897$7A08288@C47DB0F0139B24A76C02AD8FAF16871b8129d12ddc6f547e0e038f2670bbe9d6b9c6f0765f5027d667a3d3c4e979a5548c3d950c5ff43bc2043d215588ef0330b3baff5fd81b6f08fc69067f165e5
4	D4ED2DC*EC653AC04EEC99025F802E36E16276665#385C4880434CE33A6@EC95A1ED3C848253B278881%A72$85DA24D636E0FBF1F4E6F826FC956FCC37D682ef3487ed214f7f5c467195f3a4a7b90f1c9d6333bdbcb9883fcab56d97205ad7398177ce5d31e11a52b6c14a93b259109951145877101cc246456a29e47737a

The number of session IDs captured and the number of session IDs attacked are tabulated in Table.3.9

Table.3.9 Number of Session IDs captured for 248 character MAC Session ID

No	Attacks Executed	No of Session IDs captured		
		10 sessions	20 sessions	30 sessions
1	Number of unique session IDs generated	10	20	30
2	Number of session IDs captured by attacks	0	0	0
3	Number of session IDs prevented from attacks	10	20	30
4	Session Hijack Prevention Rate	100 %	100 %	100 %

3.12 PERFORMANCE ANALYSIS

The number of Session IDs prevented for 160 characters, 192 characters and 248 characters session IDs are recorded for 10 sessions, 20 sessions and 30 sessions in the Table.3.10

Table.3.10 Number of Session IDs prevented for 10, 20 and 30 Sessions

No	Length of the session ID	No of Session IDs prevented		
		10 sessions	20 sessions	30 sessions
1	160 characters	10	19	29
2	192 characters	10	20	29
3	248 characters	10	20	30

No of Session IDs attacked for 160,192, 248 MAC appended session ID is shown in the Fig.3.3

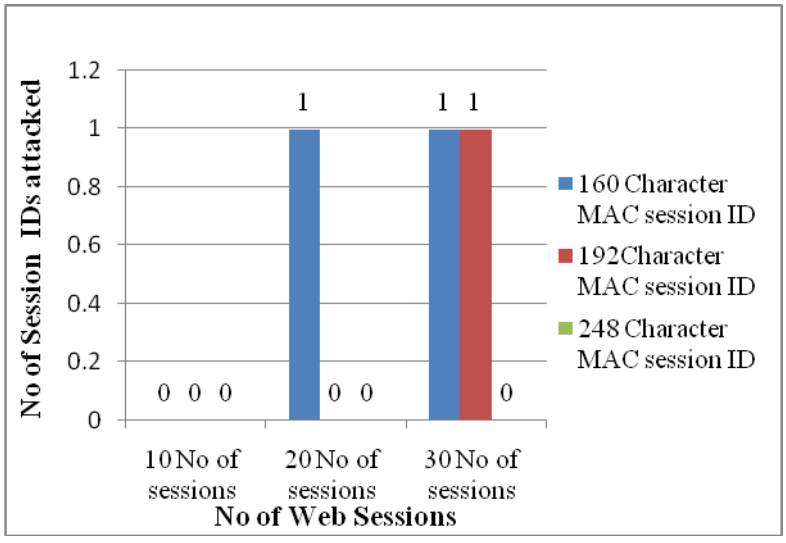

Fig.3.3 No of Session IDs attacked for 160,192, 248 MAC session ID

The number of non special characters present in the 160, 192 and 248 characters MAC appended Session ID is tabulated in the Table.3.11

Table.3.11 Number of non Special Characters in the Session ID

No	Length of the MAC appended Session ID	Number of non special characters in the session ID
1	160 character	157
2	192 character	188
3	248 character	241

The number of special characters present in the 160, 192 and 248 characters MAC appended Session ID is shown in the Fig.3.4

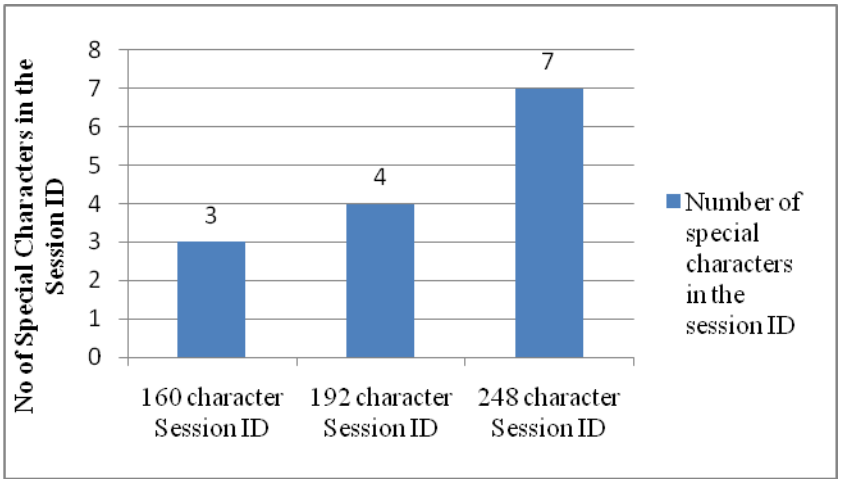

Fig.3.4 Number of Special Characters in the Session ID

The Session ID generation time for the 160, 192 and 248 characters is tabulated in the Table.3.12

Table.3.12 Session ID generation time

No	Length of the MAC appended Session ID	Session ID generation time (in milli seconds)
1	160 character	200 ms
2	192 character	240 ms
3	248 character	310 ms

3.13 COMPARISON OF EXISTING AND PROPOSED SYSTEMS

Session Hijack Prevention Rate is measured for the proposed MAC appended session ID and the prevention rate is compared with existing systems such as Unique Session ID presented by Lanxiang Chan, Dan Ferg et al. (2009), TCP state analyzer given by Bazara Barry and Anthony Chan (2007) , Received Signal Strength (RSS) discussed by Xiaobo Long and Biplab Sikdar (2010) , ARP Table discussed by Mark Lin, SANS Security Institute (2005), Secure Authentication and Key Agreement presented by Fengjiao Wang Yuqing Zhang (2008) and One way Authentication discussed by Jeffrey Cashion and Mostafa Bassiouni (2013).

The following Table.3.13 shows the comparison of session hijack prevention rate between the existing system and the proposed system.

Table.3.13 Comparison of Session Hijack Prevention Rate

S.No	Methods	Session Hijack Prevention Rate
1	Unique Session ID	85
2	TCP State Analyzer	92
3	RSS	75
4	ARP Table	87
5	SAKA	92
6	1-way authentication	94
7	**MAC appended Session ID**	**98**

The following Fig.3.5 shows the session hijack prevention rate for the MAC appended session ID.

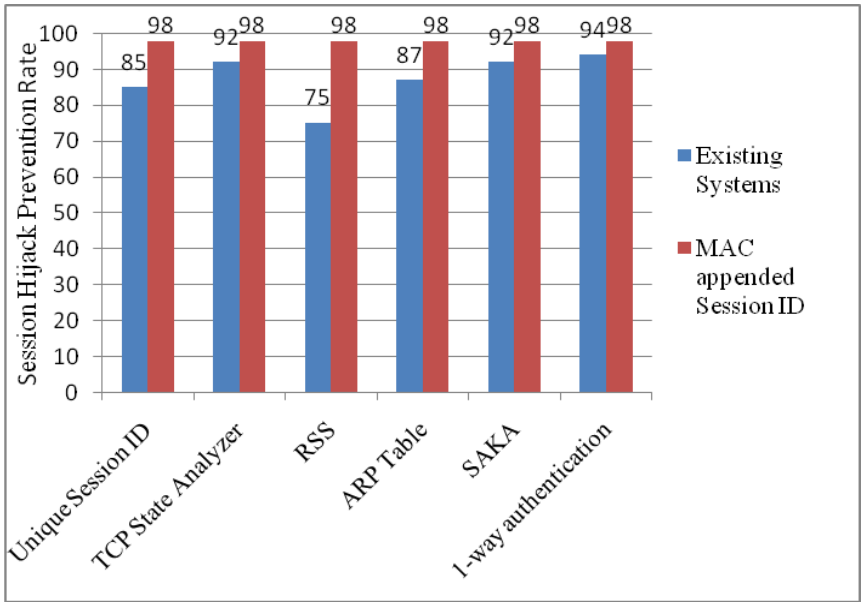

Fig.3.5 Comparison of Session Hijack Prevention Rate

Experimental results proved that Session ID length of 248 characters has 98 % prevention rate against session hijack attacks.

3.14 Discussion on the Performance of three methods

Message Authentication Code is generated by the server. The generated 128 character MAC is appended to the old session ID of 3 different lengths of 32, 64 and 120 characters. The attacks are executed to crack the session ID of 160, 192 and 248 character MAC appended session ID. For each method number of session IDs prevented is recorded for 10,20 and 30 number of sessions. 160 character MAC session ID prevents 10, 19 and 29 session IDs. 192 character MAC session ID prevents 10, 20 and 29 session IDs. 248 character MAC session ID prevents 10, 20 and 30 session IDs. The experimental results proved that 248 character MAC session ID has 98 % session hijack prevention rate. Inclusion of special characters in the MAC appended session ID is improves the session hijack prevention rate.

3.15 SUMMARY

Web applications are weakly secured against various attacks. Especially today's web applications involve with the users by establishing the sessions. Hacker's tries to steal the login credentials of users by executing the session hijack attacks in the web application. So it is necessary to secure the web applications against session hijack attacks. The proposed system analyzes the Session ID of lengths 160 characters, 192 characters and 248 characters. Message Authentication Code (MAC) is appended with the Session ID to protect the integrity of the session ID during the transmission of Session ID between client and server. The experimental results proved that MAC appended Session ID of 248 characters has 98 % prevention rate against session hijack attacks in web applications.