CHAPTER 5

Preventing the Session Hijack attacks in healthcare web applications using session key authentication and distributed session ID

5.1 INTRODUTION

Usage of health care web applications by network of hospitals and health care service providers are increases in the current technology world. Accessing the confidential healthcare information by doctors, patients over the web application is at the risk of information theft by various attacks. Most of the multispecialty hospitals situated in metropolitan cities, chief doctors are sending the prescriptions to the junior doctors over the internet after successful completion of the surgery.

Sharing the medical information with the use of latest communication technologies questions the confidentiality of the data. Most of the confidential data in health care applications are frequently hacked by powerful hackers. Deans, Doctors, patients and nurses can access the patient data over the web application. Insurance companies and health care service providers are using the same method of storing the medical information. Hackers use various hacking methods to steal the confidential data of the patient. Session hijack is a powerful attack which hijacks the session from web application users such as doctors, patients and nurses. Security requirements of healthcare applications are

(i) Confidentiality
(ii) Integrity
(iii) Availability
(iv) Privacy
(v) Authentication

5.2 HEALTH CARE SYSTEM

Current generation people requires the health care system which can be used for the patients, doctors and nurses to share the personal medical reports over online with the help of web applications.

5.3 SECURITY IN HEALTH CARE WEB APPLICATION

Healthcare web applications require strong security methods in order to prevent the attackers from stealing the patient's personal medical data. Hackers or attackers will execute the attacks to destroy the reputation of the doctor or reputation of the hospital. Attackers can use the sniffing tools to modify the confidential data transmitted between patient and doctor. It is necessary to prevent the healthcare web applications from session hijack attacks.

5.4 PROPOSED DISTRBUTED SESSION ID ARCHITECTURE

The following Fig.5.1 shows the proposed architecture for the super specialty hospital. Nowadays patients will be going to the hospitals only for the critical cases such as minor surgery or major surgery. Most of the super specialty hospitals allow the patients to communicate with the doctors with the help of hospital web application. Patients can login to their web session using http protocol and interact with the doctors in the hospitals for the diseases that can be treated with the help of tablets alone. Also the hospital web application allows the patients to send their medical reports online to the respective doctors. The following medical reports can be sent to the doctors

(i) Blood test report

(ii) Blood Pressure Chart

(iii) ECG report

(iv) Audiogram Test

(v) Scan Reports

(vi) X-Ray reports

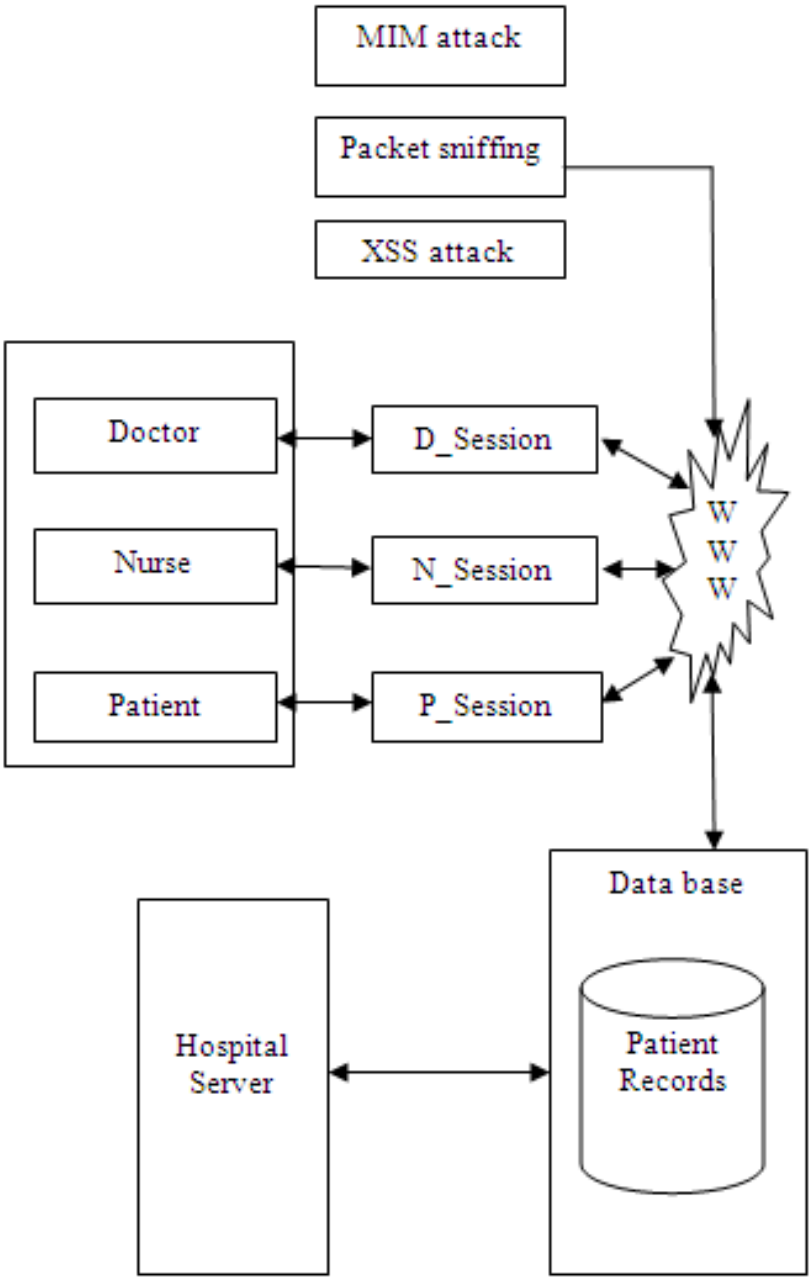

Fig.5.1 Architecture of super specialty hospital

When the patients are sending their personal medical reports to their doctor, doctor can see the reports and prescribe the medicines over online. Some of the persons who want to destroy the health of the patients, they can sniff the web application session between the patient and doctor. They can completely observe the web session content and they can alter the medicines prescribed by the doctors using the session hijack attack. Patients

think that he is interacting with doctors but actually he was interacting with attackers. These kinds of session hijack attacks are severe and can destroy the health of the patients. The proposed architecture prevents the session hijack attacks.

Patients, doctors and nurses can login to their session using their login credentials such as user id and password. In order to prevent the stealing of patient credentials by the attackers, a session key is generated using session key generation algorithm and that session key will be sent to the patient mobile number. The patient has to validate the identity by entering the session key into the hospital web application.

5.5 SESSION KEY GENERATION ALGORITHM

The idea behind the algorithm is to generate the session key (similar kind of One Time Password) to authenticate the legal client by the web server. The value of mean and variance of Gaussian Normal distribution function is chosen for the algorithm.

Algorithm :Session key generation
Input : mean , variance of Gaussian normal distribution
Output : Session Key

```
Session Key ( )
{
   Generate Gaussian Fn ( )
   {
     mean  =100f;
     variance = 0.05 f;
    }
```

Mean and variance are standard values in float as per Gaussian distribution function.

```
 float_no = d;
 int t = typecast (d);
```

Where d is the generated float value and float value is converted to integer t.

```
   v = d – t ;
```

Where v is the value obtained by subtracting t value from d value.

S_k= v * 1000000000;

}

Where S_k= Session Key

Example :

(i) The generated float value d = 782.123456789

(ii) int t = typecast (d)

t = typecast (782.123456789) = 782

(iii) v = d – t = 782 – 782.123456789 = 0.123456789

(iv) S_k = 0.123456789 * 1000000000 = 123456789

Value of the secret key S_k = 123456789

After the client is authenticated by the web server using session key, the web server splits the whole session ID into three parts and send it to the client in 3 times. Client may be patient or doctor or nurse.

5.6 NORMAL SESSION ID GENERATION

The web server generates the normal session ID using the following algorithm. Session ID is generated using Java HTTP session.

```
Generate SessionID( )
 {
    oldid = generate SessionId( );
    return sessionId;
 }
```

5.7 DISTRIBUTED SESSION ID GENERATION

The server generates the session ID and splits in to 3 parts. Client will receive the session ID in 3 parts.

Algorithm: Partitioning the Session ID

Input : Generated Session ID

Output : leftpart, midpart and rightpart of the session ID

(i) $C_m = SID_{gen} / 2$

where c_m is the middle character of SID_{gen}

(ii) If (C_m= integer)

{

integer=C_m

else

C_m= flooring(C_m)

}

(iii) $SID_{mid\_part} = \{C_{m-6}+\ldots.+C_{m-1}\}+C_m+(C_{m+1}+\ldots C_{m+6}\}$

(iv) $SID_{left\_part}= \{ C_1+C_2+\ldots+C_{m-7}\}$

where C_1 = first character of SID_{gen}

C_2= second character of SID_{gen}

(v) $SID_{right\_part}= C_{m+7}+\ldots..+ C_n$

where C_n is the last character of SID_{gen}

(vi) Server sends SID_{left_part}, $SID_{mid\text{-}part}$, SID_{right_part}

(vii) Client receives SID_{left_part}, $SID_{mid\text{-}part}$, SID_{right_part}

(viii) Client concatenates the 3 parts of SIDgen

Web server generates the session ID ranges from 36 to 68 number of characters. For example, If the length of the session ID is 44 characters, then web server splits the session ID into 3 parts such as left_part, mid_part and right_part as follows.

Length of the session ID = 44 characters

Cm = 44 / 2 = 22

Middle part of the session ID is from 16 to 28 characters. Left part of the session ID is from 1 to 15 characters and right part of the session ID is from 29 to 44 characters.

5.8 EXECUTION OF ATTACKS

Whenever the client is receiving the session ID from server, attacks such as packet sniffing, Man-in-the-Middle attack and Cross Site Scripting attacks are executed to capture the session ID.

5.8.1 PACKET SNIFFING

Wireshark tool is installed in the client machine and content of the session such as Session ID, packet header and packet data are captured.

5.8.2 MAN-IN-THE-MIDDLE ATTACK

(i) Attackers login to the session
(ii) Attacker uses sniffing tool
(iii) Attacker send the RST and FIN packet to the client
(iv) Client gets disconnected
(v) Attacker take over the session

5.8.3 CROSS SITE SCRIPTING (XSS) ATTACK

(i) create the malicious java script
(ii) inject the java script in to the web page of the created web application
(iii) steal the private data between client and web server

5. 9 RESULTS AND DISCUSSION

5.9.1 DESIGING HEALTH CARE WEB APPLICATION

In order to test the proposed system in real time the sample web application

www.chennaisuperspecialty.com is created. Separate login is created for patient, doctor and nurse. Whenever the patient or doctor enters in to their web session, web server generates the session ID and splits in to 3 parts and sends it to the client. Client integrates the session ID and communicates with the server.

5.9.2 SESSION KEY GENERATION

The following Table.5.1 shows the sample session keys generated by the session key generation algorithm.

Table.5.1 Session Keys

Sl.No	Session Number	Session Key
1	Patient Session	32267582
2	Doctor Session	24190361
3	Nurse Session	75617096

5.9.3 DISTRIBUTED SESSION ID GENERATION

The following Table.5.2 shows the generated whole session ID and 3 parts of the session ID. Web server generates the variable length session ID ranging from 36 characters to 68 characters.

Generated session ID has been spitted in to 3 parts. SID_{mid_part} will have the length of 13 characters as constant for all the sessions. But the SID_{left_part} and SID_{right_part} will have variable length of characters for each and every session based on the total length of the

session ID generated by the server. For example, a patient session can have the session ID length of 44 characters assigned by the web server.

Middle part of the session ID will have 13 length from 16-28 characters, left part of the session ID will have the length of 15 from 1-15 characters and right part of the session ID will have the length of 16 from 29-44 characters. Table.5.2 shows the sample session IDs generated.

Table.5.2 Generated Session IDs

No	Generated Session ID	SID left part	SID mid part	SID right part
1	7699812@@@@ghjlkmjhnghlerDFGK%%$$323	7699812@@@@	ghjlkmjhnghle	rDFGK%%$$323
2	88934%%%$$$@@@klmn CMNYHJREFGLMDDGH$7645678909	88934%%%$$$@@@k	lmn MNYHJREFGLMDDG	LMDDGH$7645678909

5.10 RESULTS OF ATTACKS

10 number of sessions, 20 number of sessions and 30 number of web sessions are created between the patient and doctor. The packet sniffing attack is executed using sniffer wireshark. The following Table.5.3 shows the number of session IDs captured by the packet sniffing attack.

Table.5.3 Packet Sniffing attack

No	Packet Sniffing attack	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	0	0	1
3	Number of session IDs prevented	10	20	29
4	Session Hijack Prevention Rate	100 %	100 %	97 %

The following Fig.5.2 shows the results of packet sniffing attack.

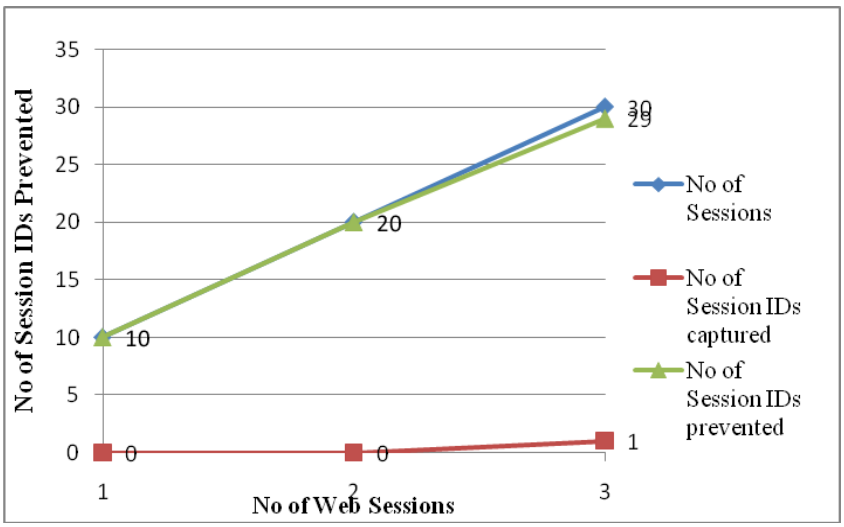

Fig.5.2 Results of Packet Sniffing attack

When the patient is interacting with doctor, Man-in-the-Middle attack is executed and the results are recorded in the Table.5.4

Table.5.4 Man-in-the-Middle attack

No	Man-in-the-Middle attack	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	0	0	1
3	Number of session IDs prevented	10	20	29
4	Session Hijack Prevention Rate	100 %	100 %	97 %

The following Fig.5.3 shows the results of Man-in-the-Middle attack.

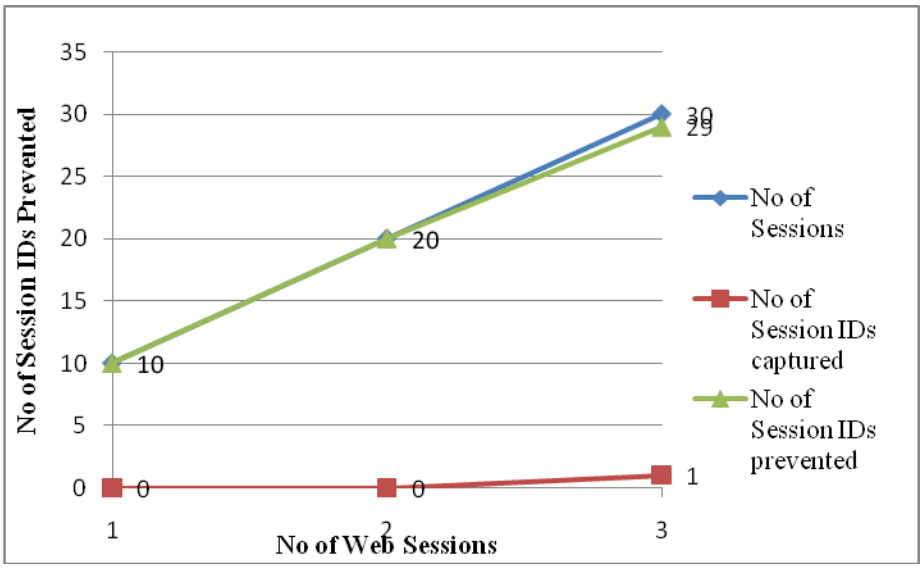

Fig.5.3 Results of Man-in-the-Middle attack

When the patient is interacting with doctor, Cross Site Scripting attack is executed and the results are recorded in the Table.5.5

Table.5.5 Results of Cross Site Scripting attack

No	Cross Site Scripting attack	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	0	0	1
3	Number of session IDs prevented	10	20	29
4	Session Hijack Prevention Rate	100 %	100%	97 %

The following Fig.5.4 shows the results of Cross Site Scripting attack.

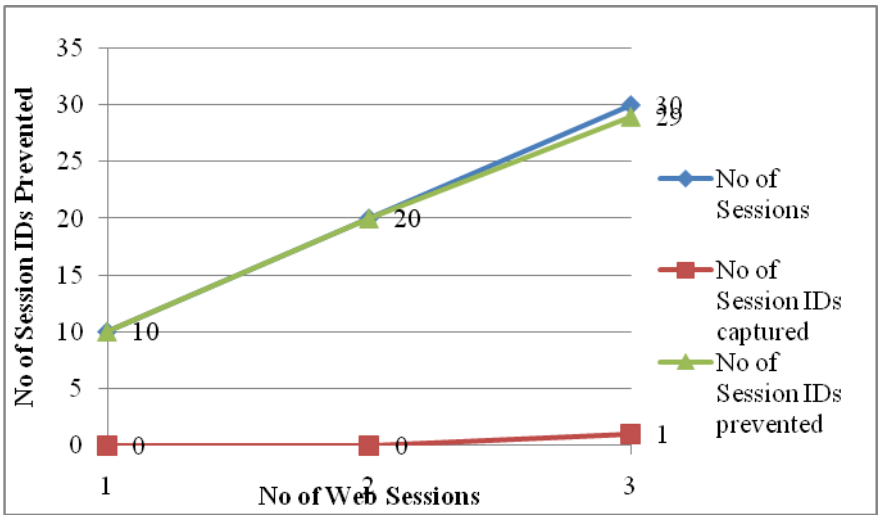

Fig.5.4 Results of Cross Site Scripting attack

5.11 COMPARISON OF DISTRIBUTED SESSION ID , MAC SESSION ID AND ENCRYPTED SESSION ID

From the results of Table.5.6, it was observed that the proposed system has the session hijack prevention rate of 99% against packet sniffing attack, 99 % against man-in-the-middle attack and 99 % against Cross Site Scripting attacks. The following Table.5.6 shows the session hijack prevention rate for packet sniffing, man-in-the-middle and Brute Force attacks for distributed session ID.

Table.5.6 Session Hijack Prevention Rate for distributed session ID

No	Name of the attack	No of Session IDs prevented		
		10 sessions	20 sessions	30 sessions
1	Packet Sniffing	10	20	29
2	Man-in-the-Middle	10	20	30
3	Brute Force attack	10	20	29
4	Session Hijack Prevention Rate	100 %	100%	99 %

The following Table.5.7 shows the session hijack prevention rate for packet sniffing, man-in-the-middle and brute force attacks for MAC appended Session ID.

Table.5.7 Session Hijack Prevention Rate for MAC appended session ID

No	Name of the attack	No of Session IDs prevented		
		10 sessions	20 sessions	30 sessions
1	Packet Sniffing	10	20	30
2	Man-in-the-Middle	10	20	29
3	Brute Force Attack	10	19	29
4	Session Hijack Prevention Rate	100 %	99 %	99 %

The following Table.5.8 shows the session hijack prevention rate for packet sniffing, man-in-the-middle and cross site Scripting attacks for Encrypted Session ID.

Table.5.8 Session Hijack Prevention Rate for Encrypted session ID

No	Name of the attack	No of Session IDs prevented		
		10 sessions	20 sessions	30 sessions
1	Packet Sniffing	10	20	30
2	Man-in-the-Middle	10	20	30
3	Brute Force attack	10	20	29
4	Session Hijack Prevention Rate	100 %	100%	99 %

The following Table.5.9 shows the session hijack prevention rate for the three methods such as distributed session ID, MAC appended session ID and Encrypted Session ID.

Table.5.9 Session Hijack Prevention Rate

No	Proposed Method	Session Hijack Prevention Rate
1	Distributed Session ID	99.7 %
2	MAC appended Session ID	99.6 %
3	Encrypted Session ID	99.8 %

The following Fig.5.5 shows the Session Hijack prevention rate between the existing systems and the distributed session ID.

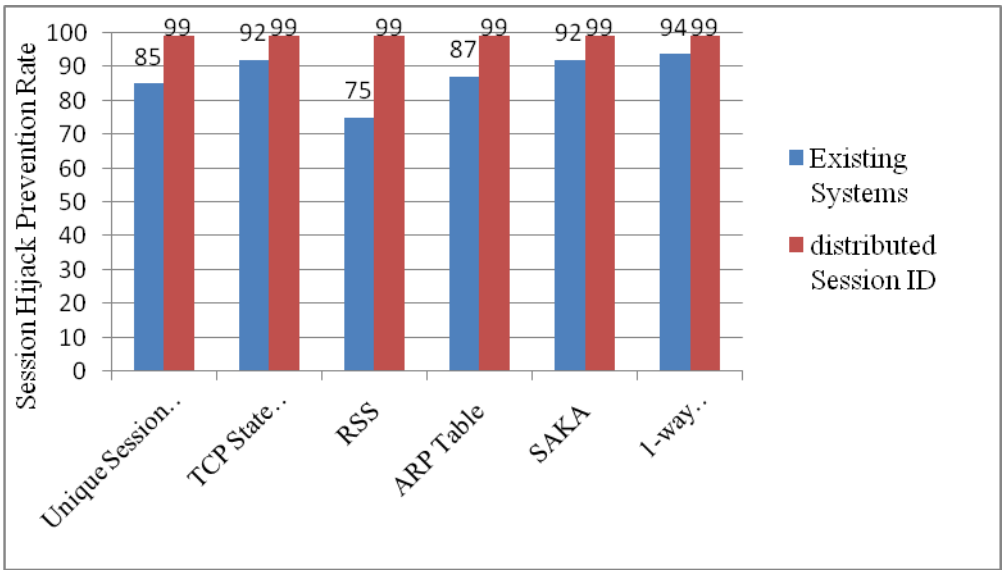

Fig.5.5 Session Hijack Prevention Rate

5.12 SUMMARY

Web application plays an important role in the field of health care. Super specialty hospitals are using the web applications to communicate with the patients. Individual web session is created for each and every time for the client. In order to prevent the stealing of sensitive medical data of patients by session hijack attacks, the strong authentication using session key is proposed for the client authentication and distributed session ID for the sessions. The experimental results proved that distributed session ID completely prevents the session hijack attacks in web applications.