Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 6, June 2016, pg.569 – 579

Man-In-The-Middle-Attack Prevention Using HTTPS and SSL

Tulika Shubh
M.Tech(CE)
tulika.shubh@gmail.com

Shweta Sharma
Assistant Professor
bhardwaj.shweta28@gmail.com

WCTM Farukh Nagar, Gurgaon, Haryana-122506

ABSTRACT: ***A man-in-the-middle attack is a type of cyber attack where a malicious actor inserts him/herself into a conversation between two parties, impersonates both parties and gains access to information that the two parties were trying to send to each other. A man-in-the-middle attack allows a malicious actor to intercept, send and receive data meant for someone else, or not meant to be sent at all, without either outside party knowing until it is too late. Man-in-the-middle attacks can be abbreviated in many ways, including MITM, Mitm, MiM or MIM.***

In this paper we will focus on attacking SSL over HTTP, known as HTTPS, because it is the most common use of SSL. We may not realize it but we probably use HTTPS daily. Most popular email services and online banking applications rely on HTTPS to ensure that communications between our web browser and their servers in encrypted. Also used combination of Diffie-Hellman and blowfish algorithm, DH for key generation and blowfish for encryption which is enhancing the data security over SSL and HTTPS.

I. INTRODUCTION

With the development of e-commerce and cloud Computing, SSL protocol is more and more widely used in all kinds of network services. SSL protocol by providing end to end authentication, message encryption, message Integrity check and other security mechanisms protects the security of the communication process. For example, Yahoo ensures the security of the mail account through SSL, to protect the safety of the user e-mail account. Amazon shields the user transaction account and transaction security by it. In recent years, due to the development of cloud computing, the connection security between the client and the cloud is also an extremely important issue. December 2009, VeriSign announced, VeriSign will provide security and authentication services cloud-based computing for Microsoft Windows Azure platform. And Microsoft will use VeriSign SSL Certificate and VeriSign code signing certificate, to ensure the security of the cloud-based computing services and applications that being developed and deployed on the Windows Azure

platform. Because, in the using of cloud computing model, the users` all computing resources are stored in the cloud, so network connection is essential for you to normally work. Therefore, if the server ends were safe enough, the security of network transmission would become very important. The SSL protocol is widely embedded in the client browser currently, the server-side also is relatively easy to deploy and implement, and the SSL protocol itself has good security features.

Man in the middle attack on SSL

A man-in-the-middle attack is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other.
Suppose Alice wishes to communicate with Bob. Meanwhile, Mallory wishes to intercept the conversation to eavesdrop and optionally to deliver a false message to Bob.

Figure MITM

The internet is growing tremendously and the data being passed is becoming crucial for the organization, we need to provide the customers as well as the organization some level of privacy and authentication so that all the users in this internet-world can assure that they are contacting the right person. To provide the user and the organization this level of security Netscape came up with a solution called Secured Socket Layer (SSL) in 1994. They never released the 1st version of SSL because of some limitations in the system . SSLv2 was released in 1994, and later IETF standardized SSL and released SSLv3.0 strengthening the protocol by adding more secured algorithms and handle the credential information more securely. They renamed the next upgraded version of the SSLv3.1 as TLSv1.0 (Transport Layer Security).

SSL was developed keeping in mind to provide information privacy and security, but still this protocol has some limitations. First of all, the SSL follows the weak model of PKI: web-model. Web-model is best for large scale implementation for the SSL, but the problem is that there are 2 types of Trust Roots in web model: CA, and User. Here user can also judge whether or not they should allow any other certifications that are not in CA. Secondly, the attacks that the SSL face are majorly from MITM attack, mainly ARP poisoning, wherein the attacker can hijack the secured session and can get the secured credentials.

In this thesis, we have discussed a scheme to strengthen the SSL using a Firefox add-on which can detect any spurious SSL certificates and a bash shell script which can be run on any Linux system to counteract against RP-poisoning.

The paper is divided in 3 parts. 1st part will discuss about general information about the SSL, about its structure and how it works. Then in the 2nd section we will be discussing about the possible attacks over SSL and how much they will be harmful. And the last section we will discuss about the solution that we have proposed.

SSL:

SSL (Secure Sockets Layer) is a standard security technology for establishing an encrypted link between a server and a client—typically a web server (website) and a browser; or a mail server and a mail client

SSL have three protocols under it: Handshake Protocol; Record Protocol; and Alert Protocol. Handshake protocol is used to establish the secure connection between the client and the server using the cipher suites and other parameters that both have agreed upon. Record Protocol is used to encrypt the data that is to be sent through the network using the key that have been established during the handshake protocol. Alert protocol is used to send the custom messages to other whenever they detect any intrusion in the system. As I need to show the defects in the SSL methods, handshake protocol need to be discussed first. It is as follows:

Step 1: Client Sends a **Client Hello** message to the server he wishes to contact. This message contains the Version No of the SSL which client can support with a 32-byte random no. this message also contains the Cipher Suites and the Compression Method that the client can support.

Step 2: Now the Server sends a **Server Hello** message to the client. This message is the complement to the Client Hello message. This message contains the version of SSL both the party will support, 32-byte random no., Session ID and the cipher suite and the compression method that it will support.

Step 3: Server then sends the **Server Key Exchange** message to the client. This message contains the public key information itself, for e.g.: the Public Key in case of RSA. Then to authenticate the client, server requests for the client's certificate information, if it has one.

Step 4: After all the information have been passed to the client, server sends a **Server Hello done** indicating the client that server's phase of initial negotiation have been done and now its clients turn.

Step 5: Now the client will send its key information to the server with **Client Key Exchange** message encrypted with the server public key so that the legitimate server only can access client's information.

Step 6: Now as both the client and the server have sent their key information and other parameters, Client sends a **Change Cipher Spec** message to the server to notify all the parameters of the secured connection and activate the same.

Step 7: Then the client sends the **Finished** message to the server to let it check the newly activated options.

Step 8: The server sends the same **Change Cipher Spec** to the client to notify all the options in the secured connections and then send the **finished** message to client to verify all the options. Next to the Handshake Protocol is the Record Layer Protocol. This layer encapsulates all the data into a frame format of size 5bytes preceding other protocol messages. This protocol provides a single frame format for Alert, Change Cipher Spec, Handshake, and Application Data.

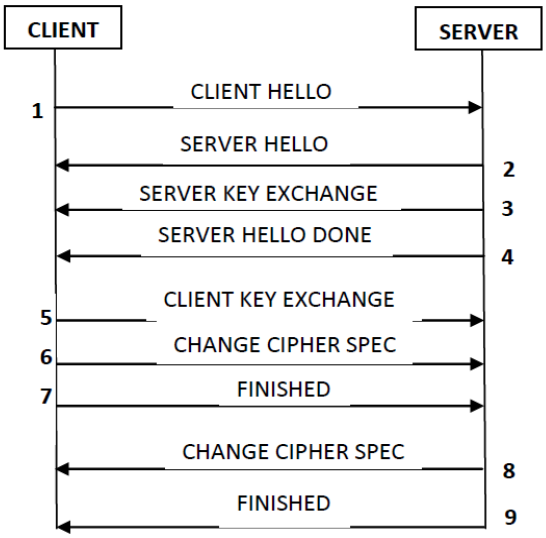


Figure 1: SSL Handshake

Protocol Overview: Blowfish Algorithm

Blowfish symmetric block cipher algorithm encrypts block data of 64-bits at a time.it will follows the feistel network and this algorithm is divided into two parts.

1. Key-expansion
2. Data Encryption

1) Key-expansion:

It will converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes. Blowfish uses large number of subkeys.

These keys are generate earlier to any data encryption or decryption.

The p-array consists of 18, 32-bit subkeys:

P1,P2,……,P18

Four 32-bit S-Boxes consists of 256 entries each:

S1,0, S1,1,……. S1,255

S2,0, S2,1,…….. S2,255

S3,0, S3,1,…….. S3,255

S4,0, S4,1,.....S4,255

In total, 521 iterations are required to generate all required subkeys. Applications can store the subkeys rather than execute this derivation process multiple times.

2) Data Encryption:

It is having a function to iterate 16 times of network. Each round consists of key-dependent permutation and a key and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookup tables for each round. Blowfish is a variable-length key, 64-bit block cipher. The algorithm consists of two parts: a key-expansion part and a data- encryption part. Key expansion converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes. Data encryption occurs via a 16-round Feistel network. Each round consists of a keydependent permutation, and a key- and data-dependent substitution.

Algorithm: Blowfish Encryption:

Divide x into two 32-bit halves: xL, xR
For i = 1to 16:
xL = XL XOR Pi
xR = F(XL) XOR xR
Swap XL and xR
Swap XL and xR (Undo the last swap.)
xR = xR XOR P17
xL = xL XOR P18
Recombine xL and xR

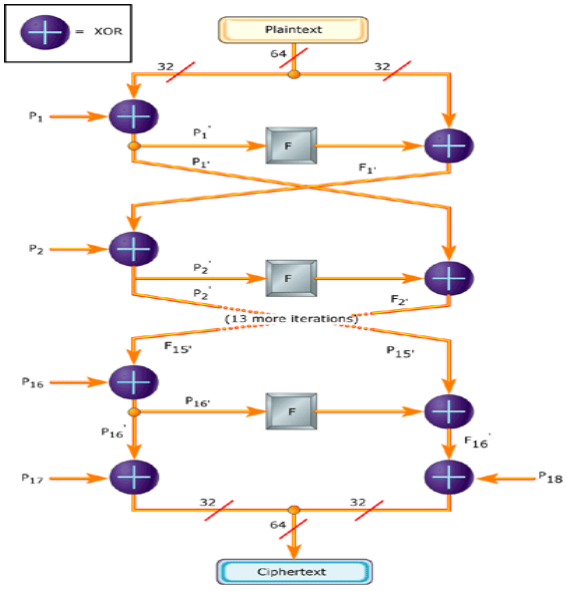


Figure1: Blowfish Encryption

Protocol Overview: Diffie Hellman

The MITM Detection Protocol is split into three stages. During the first stage, known as the Session Key Exchange stage, Alice and Bob derive a session key using the Diffie-Hellman (D-H) key exchange algorithm. Without a trusted public key exchange mechanism, the D-H key exchange is vulnerable to a MITM attack. Consequently, Alice and Bob are not certain if they are victims of a MITM attack. When Eve is present, Alice and Bob will misinterpret each other's public key with Eve's public key resulting in compromised information. Despite Eve's presence, Alice and Bob move to stage two, the Commitment stage, in an effort to detect Eve. From this point forward, information sent from Alice, Bob or Eve shall be encrypted with their derived session key. Stage two is identified as the Commitment stage where Alice forces Eve to modify her specially crafted message that contains a committed value coupled with their public cryptographic keys. If Eve decides to forward Alice's original message, then Eve's presence would be detected with certainty. If Eve desires to avoid detection, she must recreate Alice's special message coupling with Eve and Bob's public keys. However, without knowing Alice's value before commitment, Eve is forced to guess and commit a value. Alice eventually reveals her original value to Eve. To 3 maintain consistency, Eve must reveal her guessed value to Bob. Using both original values, Alice and Bob compute y. If Alice and Bob hold identical values of y, then Eve ceases to exist; otherwise, Alice and Bob are communicating through Eve. In either case, neither Alice nor Bob know if their y values match at the end of stage two. During the third stage, the Detection stage, Alice and Bob will use their y values acquired in stage two to enhance their communication environment. If Alice and Bob have identical values, then their environmental enhancements will remain consistent. However, if their values do not match, then Eve must translate messages from one communication environment to another to avoid detection. Due to physical limitations of each environment, Eve will not be able to sustain her message translation capabilities between her victims, thus being detected. There are seven scenarios that can be applied in stage three. Each scenario provides a limitation where Eve's existence is detected.

II. RELATED WORK

1. Ritesh Kumar Yadav , "Man in Middle Attack in SSL and HTTPS" International Journal of Computer Science and Mobile Computing ,Volume 4(May 2015)

With the development of e-commerce and cloud Computing, SSL protocol is more and more widely used in all kinds of network services. SSL protocol by providing end to end authentication, message encryption, message Integrity check and other security mechanisms protects the security of the communication process. For example, Yahoo ensures the security of the mail account through SSL, to protect the safety of the user e-mail account. Amazon shields the user transaction account and transaction security by it. In recent years, due to the development of cloud computing, the connection security between the client and the cloud is also an extremely important issue. December 2009, VeriSign announced, VeriSign will provide security and authentication services cloud-based computing for Microsoft Windows Azure platform. And Microsoft will use VeriSign SSL Certificate and VeriSign code signing certificate, to ensure the security of the cloud-based computing services and applications that being developed and deployed on the Windows Azure platform. Because, in the using of cloud computing model, the users` all computing resources are stored in the cloud, so network connection is essential for you to normally work. Therefore, if the server ends were safe enough, the security of network transmission would become very important. The SSL protocol is widely embedded in the client browser currently, the server-side also is relatively easy to deploy and implement, and the SSL protocol itself has good security features

2. Italo Dacosta, Mustaque Ahamad, and Patrick Traynor,"Trust No One Else: Detecting MITM Attacks Against SSL/TLS Without Third-Parties", US National Science Foundation (CAREER CNS-0952959)

The security guarantees provided by SSL/TLS depend on the correct authentication of servers through certificates signed by a trusted authority. However, as recent incidents have demonstrated, trust in these authorities is not well placed. Increasingly, certificate authorities (by coercion or compromise) have been creating forged certificates for a range of adversaries, allowing seemingly secure communications to be intercepted via man-in-the-middle (MITM) attacks.
A variety of solutions have been proposed, but their complexity and deployment costs have hindered their adoption. In this paper, They proposed Direct Validation of Certificates (DVCert), a novel protocol that, instead of relying on third-parties for certificate validation, allows domains to directly and securely vouch for their certificates using previously established user authentication credentials. By relying on a robust cryptographic construction, this relatively simple

means of enhancing server identity validation is not only efficient and comparatively easy to deploy, but it also solves other limitations of third-party solutions. There extensive experimental analysis in both desktop and mobile platforms shows that DVCert transactions require little computation time on the server (e.g., less than 1 ms) and are unlikely to degrade server performance or user experience. In short, they provide a robust and practical mechanism to enhance server authentication and protect web applications from MITM attacks against SSL/TLS.

3. Praveen Kumar Mishra ." Analysis of MITM attack in secure simple pairing" Journal of Global Research in Computer science, Volume 4 No.2 ,February 2013

A man in the middle attack is one in which the attacker intercepts messages in a public key exchange and then retransmits them, substituting his own public key for the requested one, so that the two original parties still appear to be communicating with each other. In the process, the two original parties appear to communicate normally. The message sender does not recognize that the receiver is an unknown attacker trying to access or modify the message before retransmitting to the receiver. Thus, the attacker controls the entire communication. This term is also known as a janus attack or a fire brigade attack. Active man-in-the-middle is an attack method that allows an intruder to access sensitive information by intercepting and altering communications between the user of a public network and any requested website. Avoiding logging in to sensitive sites from public locations can protect the user from conventional man-in-the-middle attacks. However, in an active MITM attack, the perpetrator manipulates communications in such a way that they can steal information for sites accessed at other times.

An active MITM may be conducted in a number of ways.

a. The attacker listens to communications transmitted over a public network.

b. The victim accesses the Internet over the network and browses to an innocuous website, such as a mainstream news site.

c. The website server processes the request and responds to it.

d. The attacker intercepts the response sent from the server and interjects an I Frame object targeting their chosen site.

e. When the user's browser receives the compromised response, it invisibly requests that website along with the cookie storing user credentials for the site.

f. This response allows the attacker to log in to the site and interact in any way that the valid user can.

4. Pushpendra Kumar Pateriya, Srijith S. Kumar, **"**Analysis on Man in the Middle Attack on SSL", International Journal of Computer Applications (0975 – 8887) Volume 45No.23, May 2012.

Man-In-The-Middle attack is the major attack on SSL. Some of the major attacks on SSL are ARP poisoning and the phishing attack. Phishing is the social engineering attack to steal the credential information from the user using either fake certificates or fake web-pages. Same in the case of ARP Poisoning, where in the attacker act as middle-man in the client-server communication channel. MITM attack makes the users difficult to understand that whether they are connected to original secured connection or not. Since the certificate that is being passed during the connection setup is insecure, attacker can easily modify the information in the certificate and leave the approval of the certificate to the user. Since many users are not well educated about the whereabouts of the forged certificates and their corresponding attacks, they accept the certificates making way for the attackers to implement the attack. To deal with such attacks, two approaches have been proposed: one for the ARP poisoning; and other for phishing attack. In this paper, Pushpendra Kumar Pateriya [3] solutions for the ARP poisoning attack and the Fake Certification Attack over SSL have been provided. The shell script will check for the ARP-IP of the gateway and other network devices and check for any modifications made into the ARP cache. The other browser plug-in may not be a successor to as that in [10], but will provide a better client side protection to the user.

5. Martin Georgiev,SubodhIyengar,Suman Jana,The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software, CCS'12, October 16–18, 2012, Raleigh, North Carolina, USA.

(Secure Sockets Layer) is the de facto standard for secure Internet communications. Security of SSL connections against an active network attacker depends on correctly validating public-key certificates presented when the connection is established. They demonstrate that SSL certificate validation is completely broken in many security-critical applications and libraries. Vulnerable software includes Amazon's EC2 Java library and all cloud clients based on it; Amazon's and PayPal's merchant SDKs responsible for transmitting payment details from e-commerce sites to payment gateways;

integrated shopping carts such as osCommerce, ZenCart, Ubercart, and PrestaShop; AdMob code used by mobile websites; Chase mobile banking and several other Android apps and libraries;
Java Web-services middleware—including Apache Axis, Axis 2, CodehausXFire, and Pusher library for Android—and all applications employing this middleware. Any SSL connection from any of these programs is insecure against a man-in-the-middle attack.
The root causes of these vulnerabilities are badly designed APIs of SSL implementations (such as JSSE, OpenSSL, and GnuTLS) and data-transport libraries (such as URL) which present developers with a confusing array of settings and options. They analyze perils and pitfalls of SSL certificate validation in software based on these APIs and present our recommendations.

6. Maryam Ahmed, BaharanSanjabi, DifoAldiaz, **"**Diffie-Hellman and Its Application in Security Protocols", International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 1, Issue 2, November 2012

With the wide use of the Internet, virtually everyone is now connected to each other through their computers. This has led to a positive impact in the human environment socially, economically and in their day-to-day transactions. There is, however, a major hindrance in trying to establish an effective and safe communication line: an outside user, not intended to be a part of the connection, might try to steal the information being passed to a legitimate user. This being a security issue, information security therefore plays a vital role in Internet transactions. It can be deduced that secure digital communication is necessary for many aspects relating to web based activities, e-commerce, and secured instant messaging. More so for private, confidential, and vital information, the reality that safe, secure communication between parties communicating over the Internet is now a necessity cannot be overstated. Cryptography is an indispensable tool for protecting information in computer systems. Today's cryptosystems are divided into two categories: symmetric and asymmetric. The difference lies in the keys used in decryption and encryption—symmetric cryptography uses the same key for both of these processes, whereas asymmetric cryptosystems use one key (the public key) to encrypt a message and a different key (the private key) to decrypt it. In this paper, Maryam Ahmed [5] The Diffie-Hellman key exchange is one of the more well-known asymmetric algorithms, formulated by its namesakes Whitfield Diffie and Martin Hellman in 1976. It is referred to in various ways, e.g. Diffie-Hellman protocol, Diffie-Hellman handshake, or Diffie-Hellman key negotiation, and commonly shortened to D–H, or DH, for convenience.

III. PROPOSED METHODLOLGY

Protecting our data online is never going to be an easy task, especially nowadays when attackers are regularly inventing some new techniques and exploits to steal our data. Here in my proposed system I had used the combination of Diffie-Hellman and blowfish algorithm, DH for key generation and blowfish for encryption which is enhancing the data security over SSL and HTTPS. The reason for selecting Blowfish is its shorter block size. And here Just Public key needs to be encrypted generated by DH. This algorithm has key expansion feature which is ideal for not taking into consideration number of transactions. Blowfish is not subject to any patents and is therefore freely available for anyone to use. This has contributed to its popularity in cryptographic software. Blowfish has a relatively large memory footprint of just over 4 kilobytes of RAM. The ultimate goal of our proposed system is to create a secure channel over insecure network.

The advantages of blowfish algorithm are:-

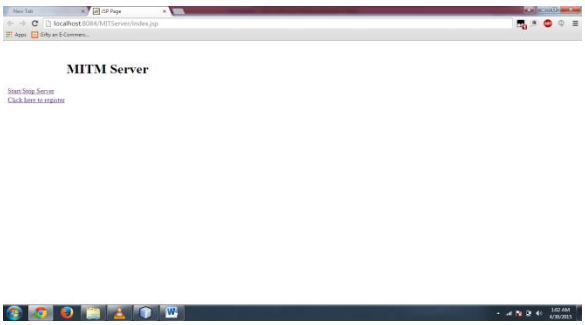
- It is secure and easy to implement and best for software implementation.
- Though it uses lot of memory and has a relatively long key setup time, but this is faster after that.

IV. RESULT ANALYSIS / IMPLEMENTATION:

4.1 Server

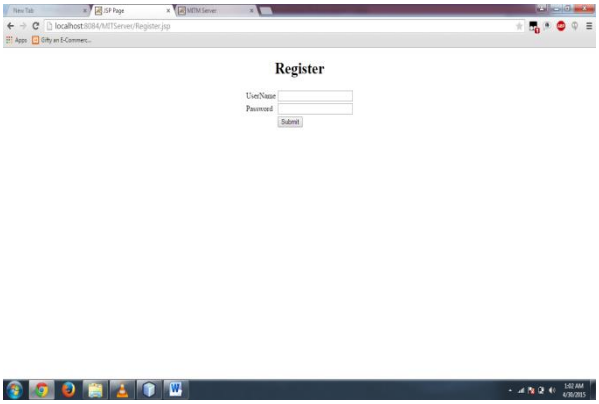
This server is made for

1. Clients registration for the first time to the bank's server
2. Detection of MITM attack over bank website



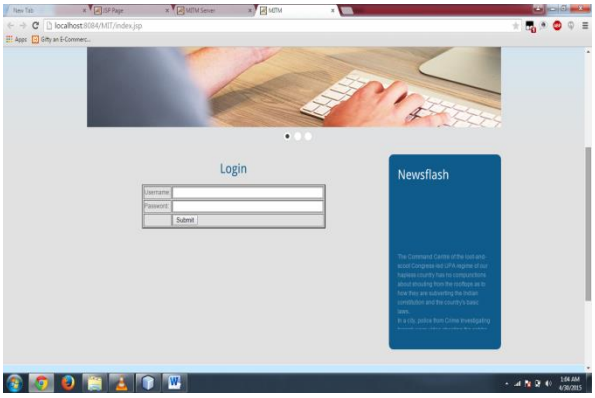
4.2 Authorization By bank's server

This is for registration of client for further communication and assignment of username and password for authorization of client on bank's website**.**



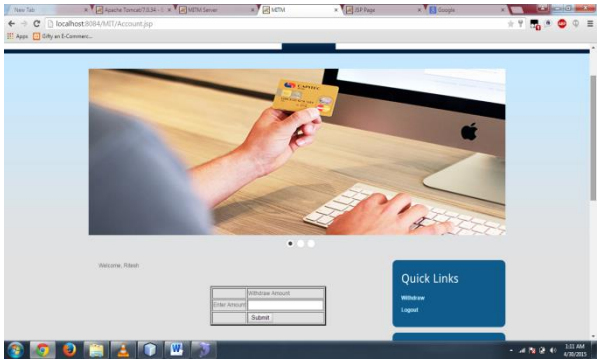
4.3 Client login

Client can login to bank's website with the username and password which is secure over bank's server.



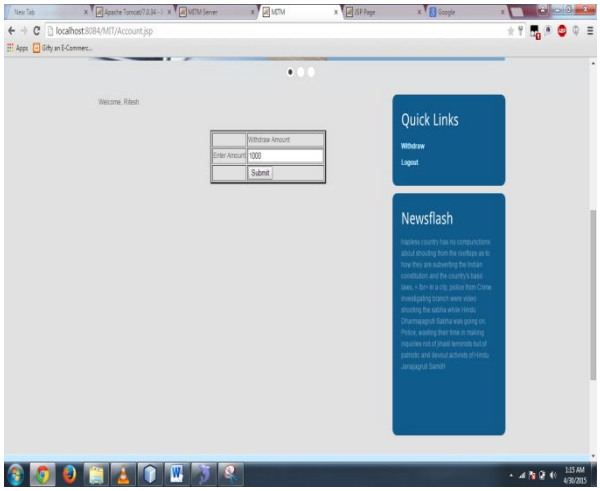
4.4 Authorized client site

This is for the users of bank's website for transaction and where they can enter the amount for withdrawal after login.



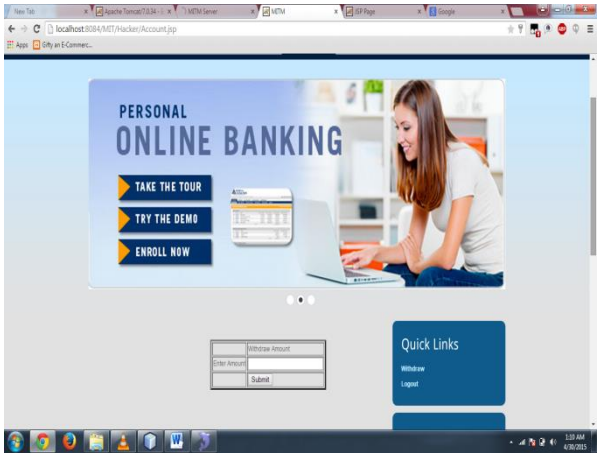
4.6 Authorized user Account

Enter Amount for transaction by authorized user.



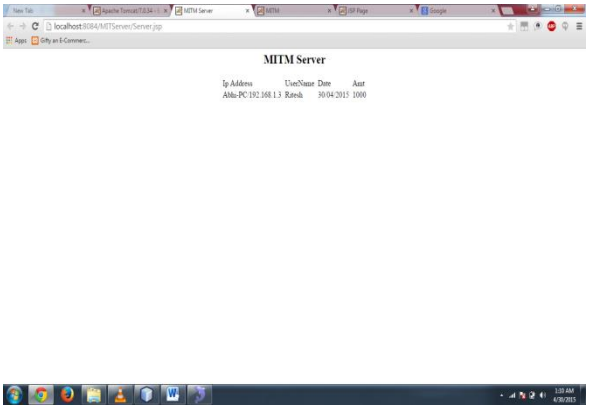
4.6 Proxy Server for Hacker

This proxy website is made same as bank's website to get the details of man in middle attacker or intruder over SSL and HTTPS.



4.7 MITM detected by server

If any MITM activity is detected by MITM server than it tracks IP address ,username, date and amount withdrawal to the server databse.



CONCLUSION

We considered Man-in-the-middle Attack while proposing the system. The middleman has traditionally been seen as evil by security protocol designers, and attempts are made to exclude him.

In our study we understood using SSL and HTTPS effectively to avoid the MITM attack. Proposed system has used the combination of Diffie-Hellman and blowfish algorithm, DH for key generation and blowfish for encryption which is enhancing the data security over SSL and HTTPS for online banking transactions.

REFERENCES

[1]. Ritesh Kumar Yadav , "Man in Middle Attack in SSL and HTTPS" International Journal of Computer Science and Mobile Computing ,Volume 4(May 2015) : 566-572

[2]. Italo Dacosta, MustaqueAhamad, and Patrick Traynor," Trust No One Else: Detecting MITM Attacks Against SSL/TLSWithout Third-Parties", US National Science Foundation (CAREER CNS-0952959): 1-16

[3]. Praveen Kumar Mishra ." Analysis of MITM attack in secure simple pairing" Journal of Global Research in Computer science , Volume 4 No.2 ,February 2013: 42-45

[4]. Pushpendra Kumar Pateriya, Srijith S. Kumar, Analysis on Man in the Middle Attack on SSL, International Journal of Computer Applications (0975 – 8887) Volume 45 No.23, May 2012: 43-46

[5]. Martin Georgiev,SubodhIyengar,Suman Jana, The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software, CCS'12, October 16–18, 2012, Raleigh, North Carolina, USA

[6]. Maryam Ahmed, BaharanSanjabi, DifoAldiaz, Diffie-Hellman and Its Application in Security Protocols, International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 1, Issue 2, November 2012: 69-73

[7] Certificate Patrol (2010), http://patrol.psyced.org/

[8] Soghoian , C ., Stamm, S.: Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL. In: Proceedings of Financial Cryptography and Data Security (2011): 1-17

[9] Parno, B., Kuo, C., Perrig, A.: foolproof Phishing Prevention. In: Proceedings of Financial Cryptography and Data Security (2006)

[10] Alicherry , M., Keromytis, A.D.: DoubleCheck: Multi-path Verification Against Man-in-the- Middle Attacks . In : Proceedings of the IEEE Symposium on Computers and Communications (2009)

[11] Thomas, S. 2000. SSL and TLS Essentials: Securing the Web. Wiley.

[12] Introduction to Secured Socket Layer. White Paper Cisco System.

[13] McKinley, H.L. 2003. SSL and TLS: A Beginners Guide. SANS Institute.

[14] Wagner, R., Bryner, J. 2006. Address Resolution Protocol Spoofing and MITM Attacks. SANS Institute.

[15] Marlinspike, M. 2009. New Tricks for Defeating SSL in Practice.In Proceedings of the Black Hat Technical Security Conference.

[16] Huawei, Z., Ruixia, L. 2009. A Scheme to Improve Security of SSL.In Proceedings of the Pacific-Asia Conference on Circuits, Communications and System, PACCS '09.

[17] Joshi, Y., Das, D., Saha, S. 2009. Mitigating Man in the Middle Attack over Secure Sockets Layer. In Proceedings of the International Conference on Internet Multimedia Services Architecture and Applications, IMSAA '09

[18] Cheng, K., Gao, M., Guo, R. 2010. Analysis and Research on HTTPS Hijacking Attacks.In Proceedings of the Second International Conference Networks Security Wireless Communications and Trusted Computing, NSWCTC '10.

[19] Jiang Du, Xinghui Li, Hua Huang. 2011. A Study of Man-in-the-Middle Attack Based on SSL Certificate Interaction. In Proceedings of the 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control, IMCCC '11

