Anonymous Authentication Against Man-In-The-Middle Attack

Mohd Izuan Mohd Saad, Kamarularifin Abd Jalil, Mazani Manaf
Faculty of Computer and Mathematical Sciences, UiTM Shah Alam, Selangor, Malaysia.
mdizuansaad@uitm.edu.my

***Abstract*—Evolving enterprise in application and data with flexible and scalable infrastructure in cloud services could improve efficiency and productivity of a business operation. Cloud services also offer resource sharing, data storage and application platform as on-demand services that could reduce the operational expenditure. Nevertheless, increasing usage and accessibility to the cloud services require strong security control to preserve user's privacy and data integrity due to network communication vulnerabilities. There are many possible attacks that could cause security breach and abuse the user's identity, leading to illegal access to the server. Man-in-the-middle attack is one of the attacks that can intercept communication between users and collect all users' information. The attacker can misuse the information and act as a legal user to gain access to the system. It is a big challenge to preserve user's privacy and provide protection from malicious attack. This paper proposes anonymous authentication scheme to preserve user's privacy and provide protection to such possible attacks. The proposed scheme also provides secure mutual authentication, anonymity, session key establishment and non-dependency with the third party. The proposed scheme uses password-based authentication as an authentication method with anonymity feature to preserve user's privacy. Experiment was conducted to test and validate the proposed scheme with man-in-the-middle attack. The result of the experiment shows that the proposed scheme is able to provide the privacy to mitigate and successfully preserve the user's identity from the attack.**

***Index Terms*—Anonymous Authentication; Vulnerability; Privacy; Man-In-The-Middle Attack.**

I. Introduction

Virtualization and less engagement on infrastructure development in cloud computing allow users to achieve better planning and operational costs. All facilities including hardware, software and licensing will be provided by cloud service provider. Cloud service provider will provide the cloud service and resources based on users demand. Flexibility and multi-tenancy characteristics are required to change current security approach. Trust and privacy are the major concerns for cloud users to adopt and migrate to the cloud environment [1]. Without proper planning and control, cloud resources are vulnerable to attacker that will cause security breach.

Vulnerability in computer system can be defined as a weakness in a system operation that allows attacker to diminish computer processing capacity. It could create such a big problem that it can devastate the entire cloud domain. Analysis on security vulnerability of system application has to be conducted to identify security loopholes in system operation. For example, clear-text username and password in authentication process should be avoided in cloud environment. Authentication process is a primary vulnerability in cloud services which can lead to high risk on data security [2]. To preserve user's privacy is a main concern to ensure that communication is trusted and secured.

There are several factors of vulnerabilities that are associated with authentication schemes [3][4]. The main concern is how secured the user's credential is protected and resistant from malicious attack. Authentication process is a frontline in communication system that will verify and identify an authorized user before they can gain access to the system. A powerful and strong authentication scheme is needed in order to provide protection against malicious attacks such as dictionary, phishing, man-in-the-middle, denial of service, impersonation, replay, stolen verifier and other feasible attacks [5].

Password-based authentication is the most popular method to authenticate users. Password-based authentication requires username and password for verification and identification of legitimate user. This is a crucial for system administrator to ensure that the user's credential is not exposed and abused in order to preserve privacy of the user. In many cases, user ID and password will be intercepted during the login process by an attacker. There are some studies that use cryptography algorithm to protect user password through encryption [6]. However, the user ID is still exposed that can lead to the attack. This paper proposes anonymous authentication scheme with adopted secure key exchange protocol called Secure Remote Password (SRP)[7]. SRP provides zero knowledge proof to preserve user ID and password from attacker. The proposed scheme also does not require any third party for verification purpose.

This paper is organized in the following sections. Section 2 presents the detail description of SRP protocol. Section 3 presents the proposed solution which introduces anonymity element in password-based authentication. Section 4 describes the preparation of solution and how the experiments are conducted. Section 5 discusses the results and analysis. Finally, Section 6 concludes the proposed scheme and future works.

II. Secure Remote Password (SRP)

In 1998 [8], Wu has introduced a very simple and efficient verifier-based protocol known as Secure Remote Password protocol (SRP-3). This protocol has been studied and improvised to achieve new optimization version and resistant from an active attack called SRP-6(2002) and later SRP-6a(2005). In this paper, SRP protocol will be used as password-based authentication which provides zero knowledge proof for verification and identification process. SRP is human-memorisable password which does not require any complex key derivation. SRP also provides mutual authentication for both parties. This feature has the advantage to establish trust relationship between both parties if the communication is in an insecure environment.

During the authentication process, SRP uses verifier value to verify and identify the legitimate user with stronger protocol flow. The password is never sent across the network and the server does not store any equivalent-password value. This feature can protect from online and offline dictionary attacks. However, the user ID is still vulnerable and it is sent across the network in plain text which exposes user identity to attacker. The attacker could attempt a guessing attack by manipulating user ID which can compromise user's privacy and data integrity. Therefore, anonymity feature is needed to hide user ID from being exposed during login transaction in order to protect user privacy and data integrity [8]. To achieve a strong, reliable and secure system architecture, the system must be resilient to such attacks and have the capability to hide identities of communication participants from third parties [3]. The use of anonymity element can improve trustworthy of the system and will eventually convince the user to use that system [10]. Thus, anonymous authentication is proposed to adopt anonymity feature in SRP. The details on proposed anonymous authentication will be explained in Section 3.

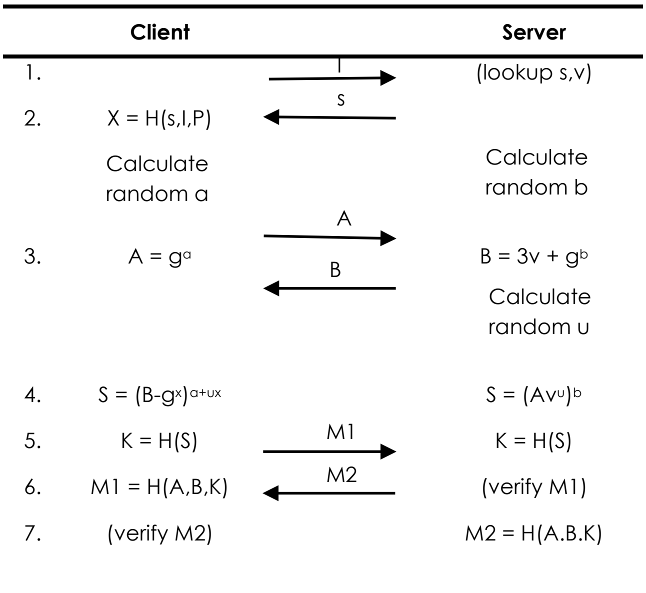


Figure 1: SRP Protocol

Figure 1 shows the step of key exchange between client and server in establishing communication session in SRP protocol. There are two phases for this protocol. The first stage is the registration phase, which is to calculate a verifier value generated from username and random salt number. Then, the client sends all the value to store in the server. The second stage is the authentication phase for verification and identification process for each step until the process is accomplished. The following is the description of SRP protocol steps [7]:

1. Client sends its username (I) to the server.
2. Server lookup client's verifier (v) and salt (s) in its database based on username (i). Then, server sends client's salt (s) to the client.
3. Upon receiving client's salt (s), client computes a private key (x) by using one-way hashing function (H). Then, client generates ephemeral private key (a) and server generates ephemeral private key (b). Both key are not publicly exposed.
4. Then, client computes a public key (A) by using generator (g) and ephemeral private key (x). At the server side, server computes public key (B) using verifier (v), generator (g) and its private key (b).
5. Client and server compute a session key (S) and then hash the session key (S) into a cryptographically strong session key (K).
6. After the session key (K) is established, client computes the first evidence message (M1) to show evidence that the client has the correct session key. Then, client sends (M1) to the server. The server verifies the M1 value by computing the evidence message (M1) and comparing it with the client's value.
7. If values match, the server calculates the second evidence message (M2) as evidence that the user also has the correct strong session key.
8. This second evidence message (M2) is sent to the client. The client calculates the second evidence message (M2) and if both values are equal, then the client will be successfully authenticated.

III. A Proposed Scheme

This section describes the details of key exchange protocol for the proposed anonymous authentication scheme. The proposed scheme will preserve user's privacy and improve trust in establishing communication between client and server. There are non-dependency of third party and provision of mutual authentication to establish the secret key session for secured communication.

A. *Protocol Parameters*

The SRP protocol computes a large set of private key shared between the two parties with large safe prime number (n). It is similar to Diffie–Hellman key exchange, which allows two parties on the same key on the network [11]. Both client and server will generate a unique shared public key, which is derived from two random numbers.

At the client's side, he/she will have a username identification (I) and password (P) for authentication. At the server's side, it has stored anonymous identity (U), which generates the client's side, verifier (v) and salt (s). All arithmetic operations are performed in module n. The values of (n) and (g) can be fixed or communicated by the server during the protocol execution. The mathematical notations used in proposed protocol are as described in Table 1.

Table 1
Mathematical Notations

Notation	Description
U	Anonymous Identification
n	A large safe prime (n = 2q+1, where q is prime). All arithmetic is done modulo n.
g	A generator modulo n (also called generator)
k	Multiplier parameter (k = H(N, g)
s	Random user's salt
I	Username
P	Password
H()	One-way hash function
u	Random scrambling parameter

a,b	Secret ephemeral values
A,B	Public ephemeral values
x	Private key (derived from P and s)
v	Password verifier
K	Cryptographically strong key session
M1, M2	Evidence messages

B. Protocol Description

The main goal of this paper is to preserve user privacy in order to establish trust in a secure communication. It also protects a user's credential vulnerability from malicious attack. In this proposed scheme, it consists of two phases: client registration phase and authentication phase.

The flow of client registration phase is shown in Figure 2. The value of (n) and (g) is assumed as known value. This phase is normally done once for each user during the account setup. In this phase, anonymous identification (U) and password verifier (v) is computed based on the Equation (1),(2) and (3). The description of this phase is as follows:

1. Client input username (I) and password (P);
2. Select salt (s) randomly;
3. Compute anonymous identity (U) by using one-way hash function (H);

$$U = H(s,I) \tag{1}$$

4. Compute password verifier (v);

$$x = H(s,I,P) \tag{2}$$
$$v=gx \tag{3}$$

5. The value of anonymous identity (U), verifier (v) and salt (s) is sent to server;
6. Server stores all the value received into its database.

The actual username and password is not sent over to the network or even stored in the database. This will preserve the user ID to be exposed during the authentication phase. Attackers face difficulty to identify the owner of the user ID before they can try to breach a password.

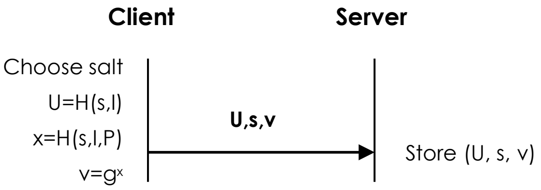


Figure 2: Client Registration Phase

Once the registration phase is completed, the client will then proceed with the authentication process by providing anonymous identifier (U). Both client and server will calculate public key (A) and (B) by choosing random ephemeral private key (a) and (b). All these value will be exchanged to calculate Session Key (S) as demonstrated in Equation (4). Finally, they will exchange evidence messages M1 and M2 to proof each other. The cryptographic strong key session (K) will be generated after the secret knowledge as Equation (5) is successfully proven.

$$\begin{aligned} S_{client} &= (B - kg^{x})^{a+ux} \quad (4)\\ &= (kv + g^{b} - kg^{x})^{a+ux} = g^{ab+uxb} = (g^{a}g^{xu})^{b}\\ &= (Av^{u})^{b} = S_{server} \end{aligned}$$

$$S_{client} = S_{server} => H(S_{client}) = H(S_{server}) = K \tag{5}$$

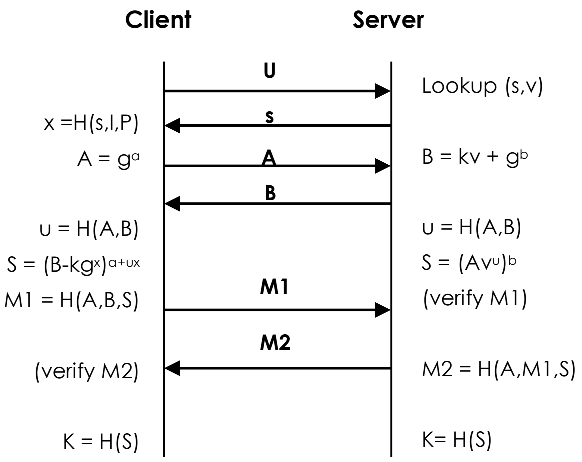


Figure 3: Client Authentication Phase

IV. EXPERIMENT

An experiment to proof that the proposed scheme is resistant from Man-in-the-Middle attack was conducted. There are three setup of machines, namely the client machine, server machine and attacker machine with different types of operating system. The client machine is a setup using Windows 7 platform installed in Intel Core i5 @ 2.4GHz with 4GB of RAM. The other two machines are setups using Linux platform (Ubuntu v14.04) with linux kernel v3.16 and install in Intel Core i5 @ 2.4GHz with 4GB of RAM. In this experiment, the client authenticates to the server via a custom web browser embedded with the proposed anonymous authentication protocol using JAVA. Client will authenticate to the server using HTTP internet browser (insecure internet protocol). HTTP is a web application protocol, which is vulnerable to session-hijacking and session-riding.

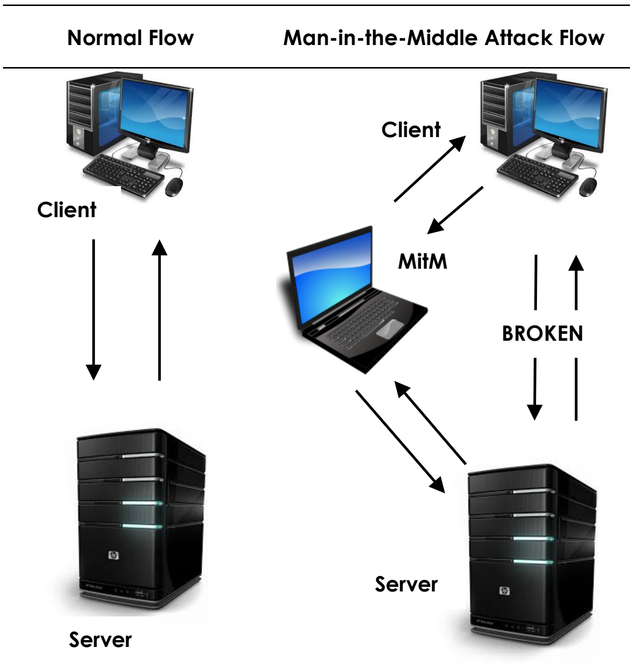
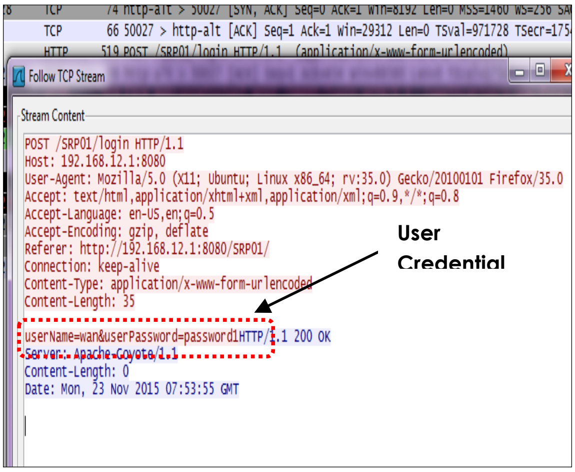


Figure 4: Diagram of Man-in-the-Middle Attack

In the attacker machine, Address Resolution Protocol (ARP) Spoofing is a setup to place an attacker machine between client and server and acts as a legal client, known as Man-in-the-Middle attack. ARP is a well-known vulnerabilities in Internet Protocols [12]. By doing ARP Spoofing, the client and server do not even notice that the traffic flow is changing through the attacker machine as shown in Figure 4. The attacker can sniff and intercept all network traffic including capturing client's username and password during the login process. For the purpose of this experiment, Wireshark [13] application was installed to monitor and capture all network packets that are transmitted between client and server.

V. Results and Discussion

The result of this study shows that the proposed mechanism was able to mitigate the Man-in-the-Middle attack in order to preserve user's privacy. Figure 5 shows that the username and password is captured by the attacker. In a normal scenario, HTTP could not protect username and password and will expose the user's credential to the attacker that leads to security breach. However, Figure 6 shows that username and password cannot be traced because it was encrypted. Although the login process is running in an insecure internet browser protocol (HTTP), it can still preserve the user's credential. Thus, the attackers cannot impersonate as legitimate users because they don not even know the real client logged in at that time. The proposed scheme shows that it can be operated in a secure and insecure environment.

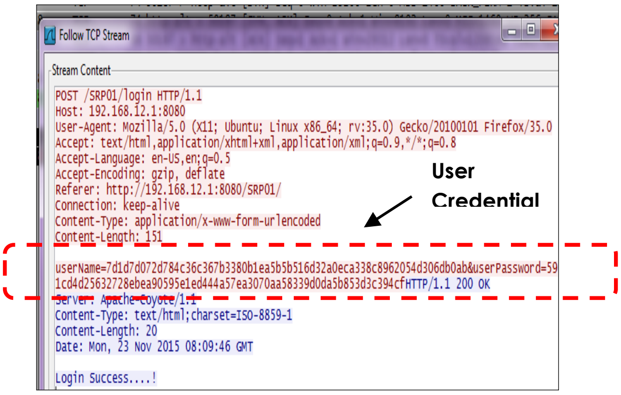


```
TCP    66 50027 > http-alt [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=971728 TSecr=175
HTTP   519 POST /SRP01/login HTTP/1.1  (application/x-www-form-urlencoded)
```

```
POST /SRP01/login HTTP/1.1
Host: 192.168.12.1:8080
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:35.0) Gecko/20100101 Firefox/35.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.12.1:8080/SRP01/
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 35

userName=wan&userPassword=password1HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Length: 0
Date: Mon, 23 Nov 2015 07:53:55 GMT
```

Figure 5: Authenticate without using the proposed scheme



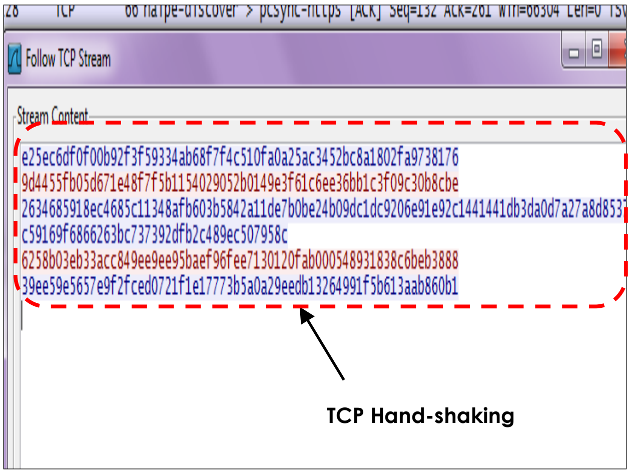
```
POST /SRP01/login HTTP/1.1
Host: 192.168.12.1:8080
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:35.0) Gecko/20100101 Firefox/35.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.12.1:8080/SRP01/
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 151

userName=7d1d7d072d784c36c367b3380b1ea5b5b516d32a0eca338c8962054d306db0ab&userPassword=59
1cd4d25632728ebea90595e1ed444a57ea3070aa58339d0da5b853d3c394cfHTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 20
Date: Mon, 23 Nov 2015 08:09:46 GMT

Login Success....!
```

Figure 6: Authenticate using the proposed scheme

Figure 7 shows the result of handshaking step in the proposed protocol. The protocol shows that each step is securely encrypted during the authentication process. In this attack, the possible way that the attacker could breach the security only if they know the secret key (k); however, it is not publicly published. Both client and server computed the key without sending it across the network. Without this key, attacker cannot compute evidence message (M1) and (M2) to establish a cryptographically strong key session (K).



```
e25ec6df0f00b92f3f59334ab68f7f4c510fa0a25ac3452bc8a1802fa9738176
9d4455fb05d671e48f7f5b1154029052b0149e3f61c6ee36bb1c3f09c30b8cbe
2634685918ec4685c11348afb603b5842a11de7b0be24b09dc1dc9206e91e92c1441441db3da0d7a27a8d853
c59169f6866263bc737392dfb2c489ec507958c
6258b03eb33acc849ee9ee95baef96fee7130120fab000548931838c6beb3888
39ee59e5657e9f2fced0721f1e17773b5a0a29eedb13264991f5b613aab860b1
```

Figure 7: TCP Hand-shaking

VI. Conclusion

Authentication is a major issue in order to achieve trustworthy in cloud services. Vulnerability in the operating system application requires new approach and method to overcome the weakness. It will lead to high risk of data integrity. Providing anonymity in an authentication scheme will hide user's credential from being exposed to any feasible attack.

This paper introduces the adoption of anonymity features to SRP protocol as anonymous authentication scheme. The main goal of this approach is to preserve user's privacy and to resist from any malicious attack. Non-dependency from third party in verification and identification process can enhance trustworthiness during communication.

Based on the experiment conducted, the result shows that the proposed mechanism can mitigate and successfully preserve the user's identity from Man-in-the-Middle attack. It also shows that attacker will not be able to impersonate the legitimate client. The proposed scheme shows that it can be deployed in an insecure network environment without any additional devices requirement.

For future work, the proposed protocol will be further study to enhance it with certificate pinning method to reduce computation overhead and roundtrips communication steps. It is a effective method to reduce latency in computation cost.

Acknowledgement

The authors would like to thank Universiti Teknologi MARA (UiTM) for their financial support in funding this research paper.

REFERENCES

[1] Sengupta, S. Kaulgud, V. and Sharma. V. S. 2011. Cloud Computing Security--Trends and Research Directions. *2011 IEEE World Congr. Serv.*524–531

[2] Yassin, A. A., Jin, H., Ibrahim, A., Qiang, W., & Zou, D. 2013. Cloud authentication based on anonymous one-time password. *In Ubiquitous Information Technologies and Applications*. 423-431.

[3] Khalid, U. Ghafoor, A. . Irum, Mand Shibli. M. A. 2013. Cloud Based Secure and Privacy Enhanced Authentication & Authorization Protocol. *Procedia Computer Science*. 22:680–688.

[4] Jaidhar, C. D. 2013. Enhanced mutual authentication scheme for cloud architecture. *Advance Computing Conference (IACC), 2013 IEEE 3rd International.* IEEE.

[5] Sood, S. K. Sarje, A. K. and Singh. K. 2009. Cryptanalysis of password authentication schemes: Current status and key issues. *Proceedings of International Conference on Methods and Models in Computer Science, ICM2CS09*.

[6] Mishra. R. 2014. Anonymous Remote User Authentication and Key Agreement for Cloud Computing. 258: 899–913.

[7] Wu. T. 2012. SRP-6: Improvement and Refinements to the Secure Remote Password Protocol.

[8] Thomas Wu. 1998. The Secure Remote Password Protocol. *Proceeding Symphosium Network Distribution System Security.98:*97–111.

[9] Goldberg, Ian, Douglas Stebila, and Berkant Ustaoglu. 2013. Anonymity and one-way authentication in key exchange protocols. *Designs, Codes and Cryptography*. 67.2: 245-269.

[10] Khattak, Z. A., Manan, J. A., & Sulaiman, S. 2011. Analysis of Open Environment Sign-in Schemes-Privacy Enhanced & Trustworthy Approach. *Journal of Advances in Information Technology*. 2:109–121.

[11] Jung, Kyoung-sook, Ji-young Kim, and Taechoong Chung. 2003. Password-based independent authentication and key exchange protocol. *Proceedings of the 2003 Joint Conference of the Fourth International Conference*. Vol. 3. IEEE.

[12] Modi, C. Patel, D Borisaniya,. B. A. Patel, and M. Rajarajan. 2012. A survey on security issues and solutions at different layers of Cloud computing. *Journal Supercomputer*. 63:561–592.

[13] Wireshark. From : http://wireshark.sourgeforge.net/ [Accessed on 10 October 2015].