Volume 7 Issue 2 (2019) 94-97 ISSN 2347 - 3258
International Journal of Advance Research and Innovation

Live Monitoring For Forensic Artifacts from IM Messenger Packets Using Freeware

Sankarshana Kadambari[1], Bhupendra Singh Chauhan[2], Mohit Soni[2]
[1]Department of Computer Science, ASET, Amity University Haryana, India
[2]Division of Research and Development, Lovely Professional University, Punjab India

Article Info
Article history:
Received 10 January 2019
Received in revised form
20 March 2019
Accepted 28 April 2019
Available online 15 June 2019

Keywords: Wireshark, Network Forensics, SnapChat, Controlled environment, IM Packets, QUIC, STUN

Abstract

Numerous smartphone applications such as snapchat pose a major problem for a network administrator, as the chat gets deleted automatically removing every evidence of a conversation. It becomes difficult for an administrator to confirm whereabouts of a captured packet belonging to an IM application. However, if the same is captured in real time using Wireshark-a detailed analysis of the protocols would reveal information regarding the source of packet generation. This paper emulates a closed environment and uses freeware to capture encrypted packets from instant messengers and attempts to produce sufficient artifacts, so as to pin point the sender.

1. Introduction

Network or protocol analyzer is a program that runs on a device that is connected to the network, it passively receives all data link layer frames passing through the device's network adapter. The analyzer captures the data that is addressed to other machines, saving it for later examination [1]. One of the freeware used in the research is]Wireshark. It is having an interactive GUI which displays all the packets in order, it has many filters available which are in the form of protocols. There are color codes present for various protocols such as green for TCP packets, dark blue for DNS packets, light blue for UDP packets, and black identifies TCP packets with glitches. In this paper, Wireshark observes traffic that passes through mobile hotspot created on laptop and the packets which belong the concerned application are sorted out. Then the relevant information is analyzed with the help of protocols.

Instant Messaging applications are commonly used by wide range of Internet users. These Instant messaging applications are also used in Smartphones these days, they are known as Apps. Any data that travels in a form of packets over a network can be viewed using Network Protocol Analyzer and they can be recorded, monitored also in some cases read. The recorded data is used lawfully by a network administrator to monitor and troubleshoot network traffic. Using the information captured by the freeware an administrator can identify inaccurate packets[1].

Any traffic analysis can be classified into three types: real-time analysis, batched analysis and forensics analysis [2].

i) Real-time analysis: It is performed on data that is obtained or using small batches also known as buffers to efficiently analyze data. The response time of this kind of analysis is understood by time elapsed which is either computed or detected. Real time analysis has generally high computational resources requirements. (2)Batched analysis: Batched analysis performs analysis periodically, where the period is enough to collect data in also known as data batches. Depending on the batching policies the response time and related computational resources requirements may be higher or lower, but in general they propose a higher response time and lower computational resources necessities than real -time examination (although they require larger storage size).

ii) Forensics analysis: Forensics analysis is analysis done when a certain event occurs. An example of forensics analysis is the investigation performed when an intrusion is noticed to a host who is associated to the network. This kind of analysis require that data had been previously stored to be analyzed, and may also require of human intervention. Network data examination techniques obtain information of network data by inspecting network header fields of each packet, calculate them and produce outcomes or results. Packet in which packets are decoded and presented in a comprehensible way. Network analyzers like tcpdump, Wireshark are some examples of packet Interpreting applications[2]. In this paper, forensic analysis has been performed and forensics

1) Sender information.
2) Time stamps of packets.
3) Important protocols such as handshake and their timestamps
4) artifacts which are considered here are: -

Corresponding Author,
E-mail address: mohit.soni@outlook.com

2. Network Analysis Experimental Setup

To intercept the network traffic, wireless access point was created to which both mobile devices were connected. This was established using the Windows 10 Virtual Wi-Fi Miniport Adapter feature. This feature enables users to create a virtual network that perform as a wireless access point for numerous devices. To do this, the host computer was linked to the Internet via an Ethernet cable so that the wireless card was not in use. The Ethernet connection was established to part its Internet access with the virtual Wi-Fi Miniport Adapter. Command netsh wlan set hosted network mode = allow ssid test key = 1234567890 was executed to setup the virtual network. The network was then enabled using the command netsh wlan start hosted network. Next, the network traffic was recorded to and from the mobile devices by capturing data sent over the virtual connection. The number of packets dropped and the capture rate were not recorded, as it was not relevant to the goal of this research. Wireshark was used to capture and analyze the network traffic[3]. This set up is shown in Fig.1.

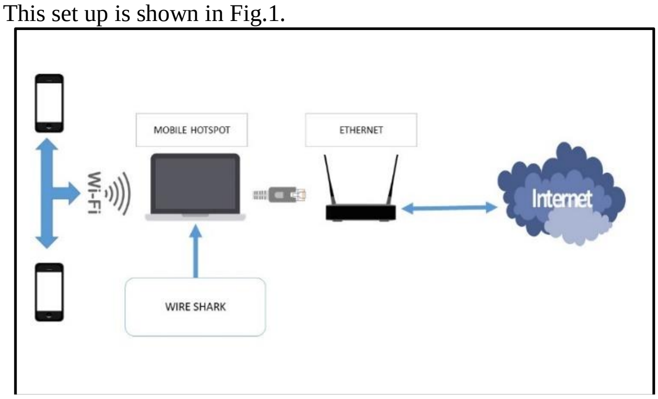

Fig.1: Setup for Capture

3.Emulation of the Problem

Initially in the procedure we must analyze the recorded data for the registered mac address. Let us consider 2 devices namely "A" & "B", their Wi-Fi Mac address and the consumed bytes are mentioned in the figure 2.

Now the second step involves the identification of server which can also be recognized with the help of endpoints feature in Wireshark and we can further confirm it from web as shown in Fig.3 & Fig.4. Fig.5 displays the SYN packet and & Fig.6 displays [SYN, ACK] packet with timestamp.

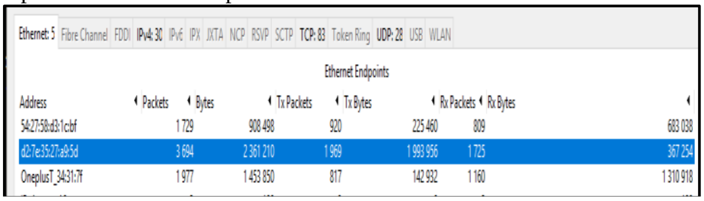

Fig.2: Endpoints of MAC

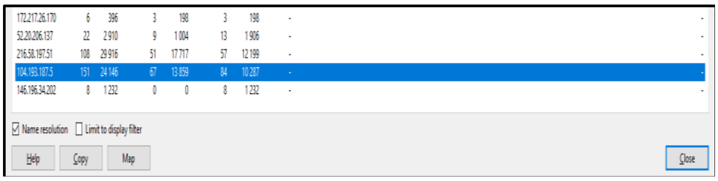

Fig.3: Server Address

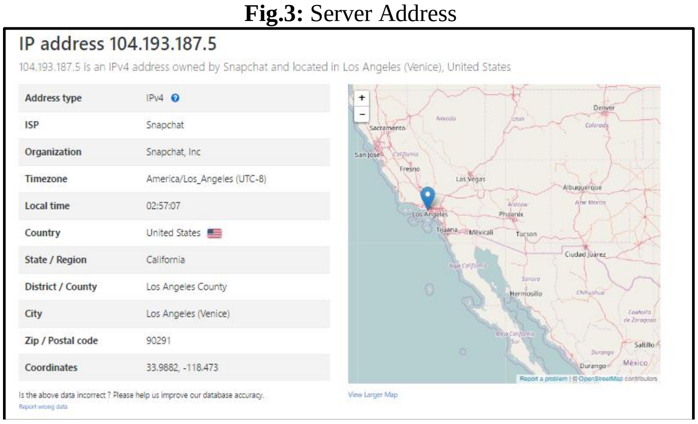

Fig.4: Server Address Confirmed

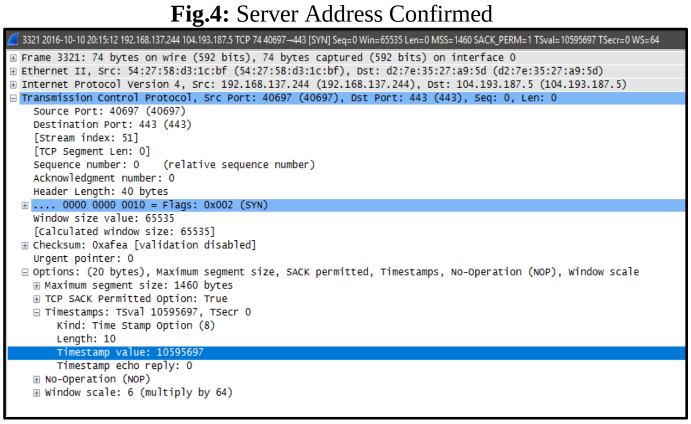

Fig.5 Packet Sent by "A".

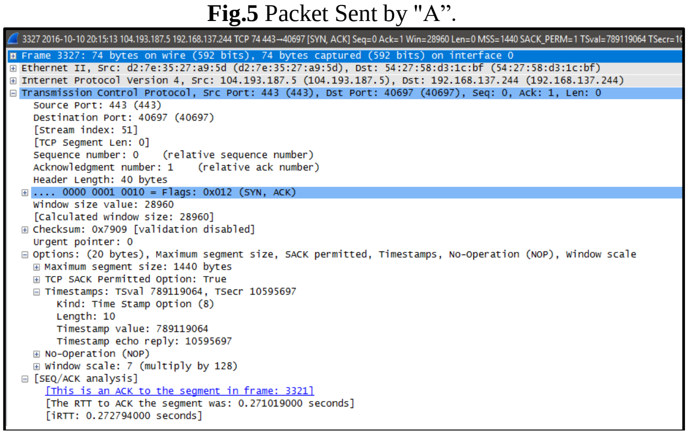

Fig.6: ACK Packet Sent by Server

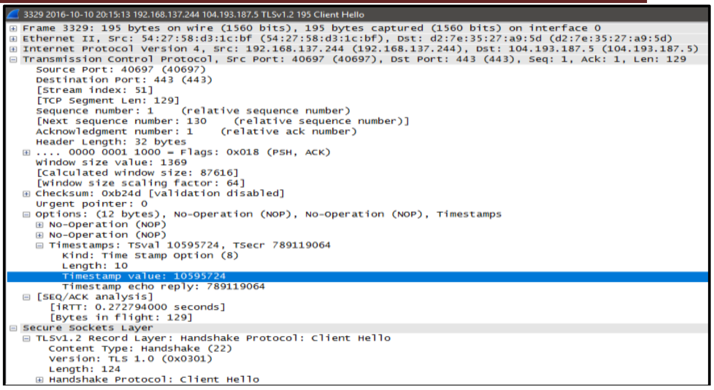

Fig.7: Client Hello by "A"

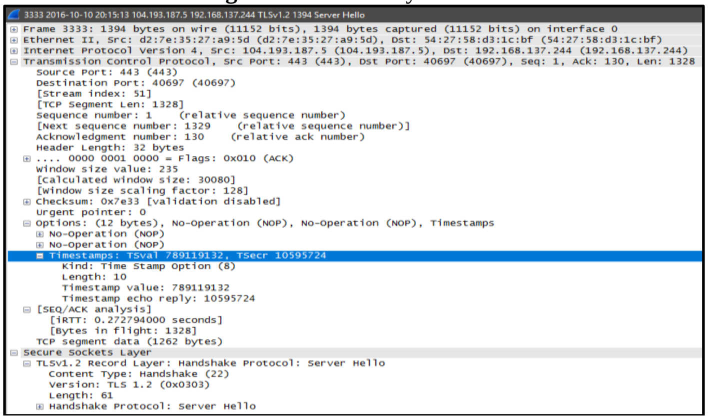

Fig.8: Server Hello

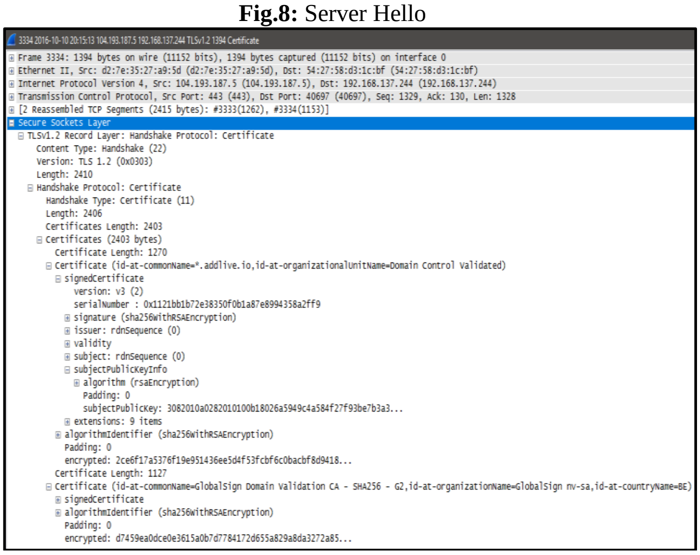

Fig.9: Certificate Sent by Server

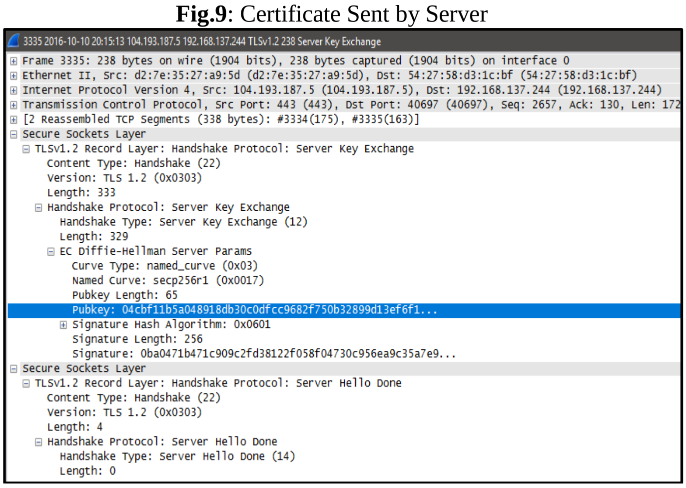

Fig.10: Server Key Exchange

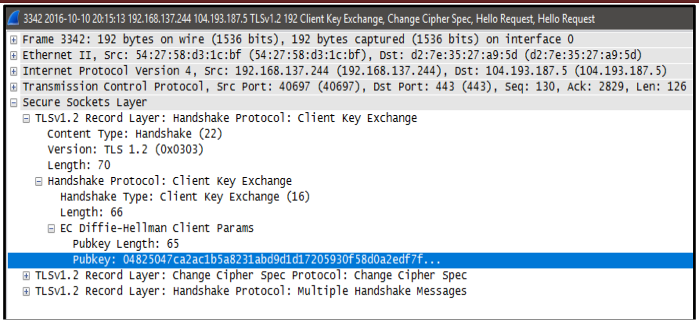

Fig.. 11: Client Key Exchange, Change Cipher Text

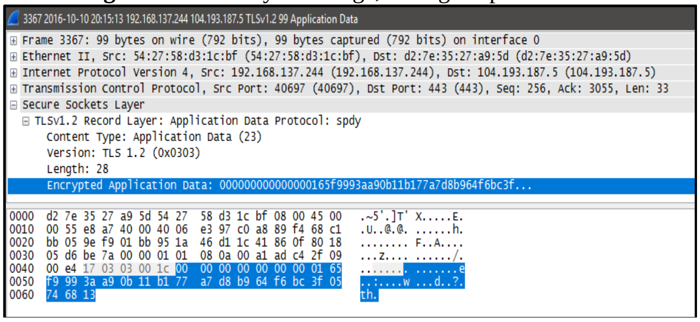

Fig.12: Application Data Sample 1

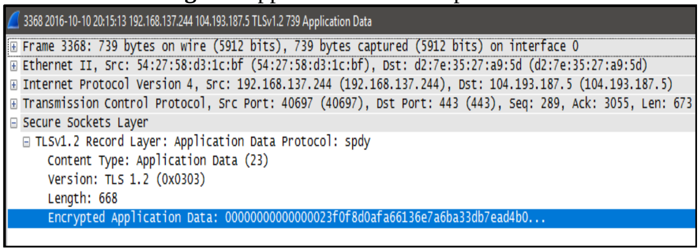

Fig.13: Application Data Sample 2

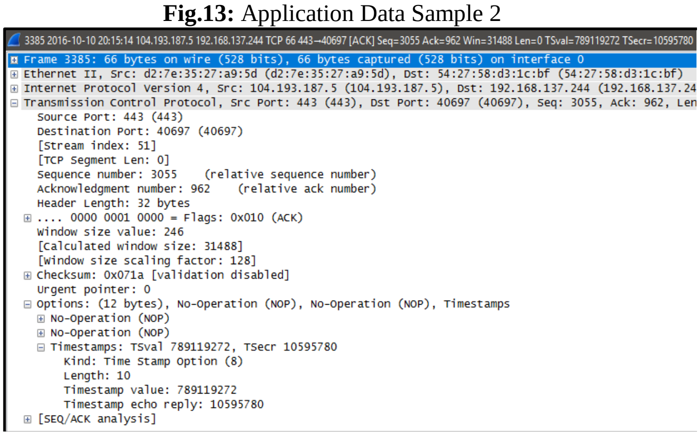

Fig.14:ACK for Application Data

After analyzing A's initial activity there were also binding request made by A shown in Fig. **15**.

4. Inference

As discussed about there are two devices namely A and B in a controlled environment where Ip address of A=192.168.137.244 ,Ip address of B is 192.168.137.244.From Fig.3 &Fig.4 it is confirmed that the server address is 104.193.187.5.The first device to initiate the connection with server is A as seen in Fig.5 and it establishes a 3-way TCP handshake which consist of SYN,SYN-ACK,ACK the time stamp value for ACK are in Fig.6 . The TCP handshaking mechanism is designed so that two computers/devices trying to communicate can negotiate the parameters of the network TCP socket connection before transmitting data[4].

The first message is the Client Hello. Since the client machine is requesting the secure communication session, this message involves a set of choices that the client is willing to use to communicate with the server. The option categories are: Version of SSL to be used, Cipher Suites supported by the client, and Compression Methods used by the client. Other information that is included in this message is a 32-byte Random Number that assists the client in establishing encrypted communications, and a SessionID field that is blank (5).

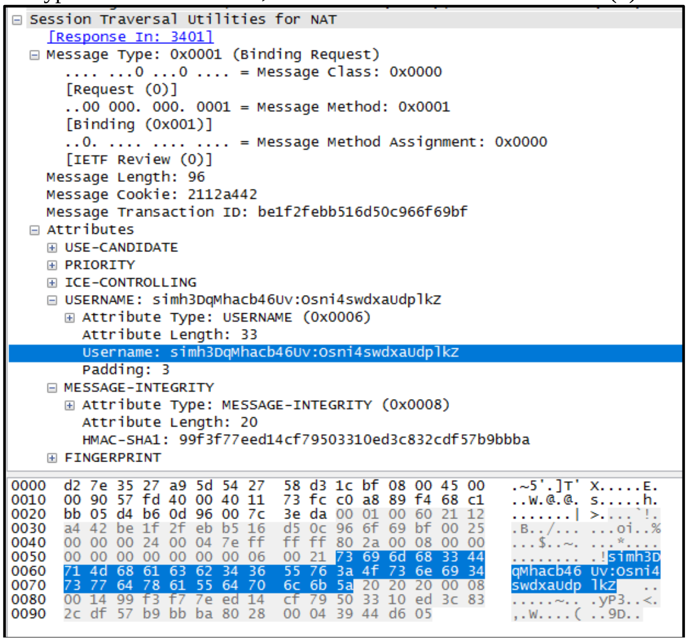

Fig. 15: Binding Request Made by A

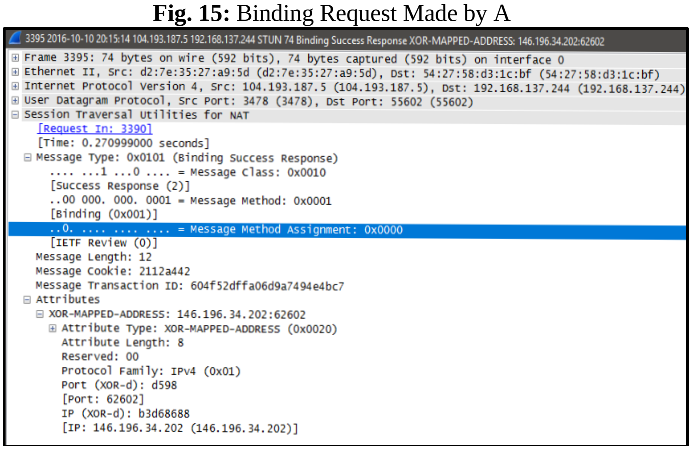

Fig.16: Binding Response by Server

The second message consist of SSL handshake is the Server Hello. In this message, the server makes choices based on the Client Hello message and makes firm decisions on the Version of SSL to be used, the Compression Method and Cipher Suite [5].

Similarly, A also generates Certificate and agrees with server how data will be encrypted as shown in Fig.9 ,the screenshot also has the public key and the certificate can also be extracted on the local pc. After this A generates ClientKeyExchange message contains information about the key that the client and server will use to communicate. This is the point where man in the middle can be performed [6].

Fig.12 and Fig.13 shows the sample application data sent by A to SnapChat server and the acknowledgment received from the server Fig.14. Device A also makes binding request to the Snapchat server using STUN protocol. One of the two major fields in STUN is Username and Message Integrity. The USERNAME attribute is used for message integrity. It identifies the username and password combination used in the message-integrity check. The value of USERNAME is a variable-length value. It MUST contain a UTF-8 [RFC3629] encoded sequence of less than 513 bytes, and MUST have been processed using SASL prep (RFC4013)[7]. In our case A's username is "simh3DqMhacb46Uv:Osni4swdxaUdplkZ". The MESSAGE-INTEGRITY attribute contains an HMAC-SHA1 [RFC2104] of the STUN message. The MESSAGE-INTEGRITY attribute can be present in any STUN message type. Since it uses the SHA1 hash, the HMAC will be 20 bytes. The text used as input to HMAC is the STUN message, including the header, up to and including the attribute preceding the MESSAGE-INTEGRITY attribute. The key for the HMAC depends on whether long-term or short-term credentials are in use. For long-term credentials, the key is 16 bytes [7]:

key = MD5(username ":" realm ":" SASLprep(password))

So, A makes a binding request with user name and server of SnapChat replies with Binding Success Response XOR-MAPPED-ADDRESS: 146.196.34.202:62601. The XOR-MAPPED-ADDRESS attribute is reflexive transport address is obfuscated through the XOR function [8-9].The same process is repeated by device B and the exchange through a common server of SnapChat with IP: 104.193.187.5. This gives enough evidence with artifacts that which Server was A in communication with and parallel to that what was the corresponding device.

5. Conclusions

Sufficient artifacts have been collected to pin point the sender from Wireshark. The two main protocols Tls1.2v and STUN have also been covered.

To decrypt a 2048-bit RSA TLS cipher text, an attacker must observe 1,000 TLS handshakes, initiate 40,000 SSLv2 connections, and perform 2^50 offline work. The victim client never initiates SSLv2 connections. An implementation of the attack and that can decrypt a TLS 1.2 handshake using 2048-bit RSA in under 8 hours, at a cost of $440 on Amazon EC2. Using Internet-wide scans it is found that 33% of all HTTPS servers and 22% of those with browser-trusted certificates are vulnerable to this protocol-level attack due to widespread key and certificate reuse. Given an unpatched SSLv2 server to use as an oracle, TLS cipher text can be decrypted in one minute on a single CPU—fast enough to enable man-in-the-middle attacks against modern browsers/applications. Procedures are easily available on web for initiating an attack like this.

A solution to this problem might be Quick UDP Internet Connection (QUIC). As TLS and its security model use one session key, while QUIC uses two, and the data may start being encrypted before the final session key is set. Second, QUIC does not run on top of TCP and implements many of the features provided by TCP itself. This is done primarily for performance reasons, but QUIC also adds some cryptographic protection, such as protection against IP spoofing and packet re-ordering.

References

[1]. Arshad Iqbal,et al. Network Traffic Analysis and Intrusion Detection using Packet Sniffer. Second International Conference on Communication Software and Networks 2010,.

[2]. P Asrodia, H Patel. Network Traffic analysis using Packet Sniffer. International Journal of Engineering Research, 2, 2012, 3.

[3]. D Walnycky, I Baggili, A Marrington, J Moore. Network and device forensic analysis of Android. The International Journal of Digital Forensics & Incident Response, 2015, 8.

[4]. InetDaemon. TCP 3-Way Handshake (SYN,SYN-ACK,ACK). http://www.inetdaemon.com/tutorials/internet/tcp/3-way_handshake.shtml.

[5]. Sans. Ssl And Tls :Beginner's Guide. 2013. https://www.sans.org/reading-room/whitepapers/protocols/ssl-tls-beginners-guide-1029.

[6]. S A Thomas. SSL and TLS Essentials: Securing the Web. New York : Wiley Computer Publishing.

[7]. J. Rosenberg, R. Mahy,P. Matthews. Proposed Standard. 10, 2008. https://tools.ietf.org/html/rfc5389#page-34.

[8]. N Aviram, S Schinzel, J Somorovsky, N Heninger, M Dankel. Drown: Breaking TLS using SSLv2. 2016. 25th Usenix Security Symposium. p. 18.

[9]. Rt Lychev, S Jeroy, A Boldyrevaz. How Secure and Quick is QUIC? 2015. IEEE Symposium on Security and Privacy.