

NETWORK TRAFFIC BASED ANALYSIS OF SECURE COMMUNICATION APPLICATION (LINE)

MSc Internship Cyber Security

Padmaja Sekher Student ID: X17135885

School of Computing National College of Ireland

Supervisor: Imran Khan

National College of Ireland



MSc Project Submission Sheet

School of Computing

Student Name	:	Padmaja Sekher	
Student ID	:	X17135885	
Programme	:	Cyber Security	Year: 2019
Module	:	Internship	
Supervisor	:	Imran Khan	
Submission Due Date	:	12/12/2019	
Project Title:		twork Traffic based Analysis of Secure C plication (LINE)	ommunication
Word Count:		Page	
word Count:	CC	ount	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

<u>ALL</u> internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature:

Date:

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple	
copies) Attach a Moodle submission receipt of the online project	
submission, to each project (including multiple copies).	
You must ensure that you retain a HARD COPY of the project, both	
for your own reference and in case a project is lost or mislaid. It is not	
sufficient to keep a copy on computer.	

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only

Signature:	
Date:	

Penalty Applied (if applicable):	

Network Traffic based Analysis of Secure Communication Application (Line)

Padmaja Sekher X17135885

Abstract

The expanded utilization of social media has an effect in everyday life exercises. Because of the expanded use of texting applications, individuals' correspondences are changed in a totally extraordinary way. As the worry for individual's security began to expand, the application suppliers began to build their applications with protection highlights. These overhauls empower the individuals to convey in a safe way thus does the lawbreakers. Despite the fact that the exchanging the messages are being encoded, breaking down the system traffic of those correspondences can uncover a few intriguing data. In this paper, the protected system traffic of secure texting application LINE was investigated and the discoveries of data about included text message, voice/video call, client exercises and different examples about the application's use were introduced. [13]. These curios can be utilized in criminal examination in the realm of cybercrime and furnish the agent with data about the application client and their exercises.

1 Introduction

The cell phones have become irreplaceable part of the human life. Nearly everybody has one cell phone. With the progression of web, individuals began to discuss through the web. Some social media administrations and their texting applications give individuals numerous selections of correspondences from messages, voice calls to video calls. With the data privacy aspects among the clients has been increased, the secure information of an application turns into its one of a kind selling point. Numerous application suppliers began to give end to end encryption and this makes the entrance to access the content by an outsider troublesome. These protected highlights additionally began to pull in offenders. The criminal investigations are getting intricate if these protected applications are being encoded by encryption it's very difficult to analysis the data content from the encrypted network traffic. Henceforth the forensic network began to consider and investigate these secure applications which has encrypted connection and attempting to discover measurable artefacts that are left by those applications. The investigation of secure applications are of two sorts. One is breaking down the artefacts left by the applications on the host gadget for example smartphones. Second is examining the artefacts that can be found in their network traffic. The network traffic can be investigated for security vulnerabilities in the conventions utilized by the applications and for designs that are distinguished which can be utilized to extricate data about the clients and their exercises or following up the activities in the application around the world. In this paper, I have presented the detailed analysis of the network traffic behavior of the secure communication application (LINE) messaging application which has end to end encryption technique. Therefore, presented the findings of the data information regarding the application users and the involved parties. Hence, discussed about how the end to end encryption application leaves the artefacts in the system by analysis the network traffic between the involved parties. LINE is one of the widespread messaging applications in the world. It has more than 500 million users downloading this application in the play store and

has a user count more than 700 million users in the worldwide. Line has a cross platform support and generally underpins around six stages like an android, IOS, BlackBerry, Firefox OS, Windows and mac OS. It is likewise one of the few messaging applications which rely on end to end encryption technique. With this sort of huge client base, the need to profile the client exercises or activities is significant, since the data contents can't be accessed by the third party which is ended to end encrypted by encryption techniques. The discoveries introduced in this paper will help the examiners who are exploring the cases which includes the utilization of the LINE delegate by lawbreakers/targets/criminals. All the more accurately, the commitments of this paper can be condensed as pursues:

- Analysis of the network traffic of LINE messaging application with the protected correspondence is examined for different client exercises or activities and their specific unique signatures are analyzed.
- Network traffic analysis is done for various operation system platforms have been discussed and profiling for client exercises/ user activities with unique signature is done.

The remaining of the paper is sorted out as pursues: In segment related works, inspecting of related existing work is detailed. In area Research methodology, the devices and the approach utilized for the examination is talked about. At this point, in specification segment it explained about the framework for analysis of LINE secure traffic is done. In section implementation, the analysis of LINE network traffic is analyzed in different platforms.

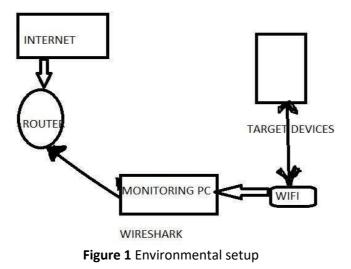
2 Related Work

Most of the popular instant messaging applications like WhatsApp, Telegram, Viber, LINE etc., have started to provide end-to-end encryption. The increase in the usage of these secure instant messaging apps made the forensic community to focus much on the artefacts or traces left by those secure apps. Many researchers worked on the artefacts that can be retrieved from the host smartphones which are left by the usage of those applications. Fazeel Ali Awan (2015) [1] and F. Norouzizadeh Dezfoul et al (2016) [2] explained the forensic artefacts that can be retrieved from the smartphones where the applications were used. They used the database files present in the image of the smartphone and retrieved the contacts, call history and duration etc. But the network artefacts that are left by those secure apps are being studied sparsely. Before the end-to-end encryption features, the traffic can be intercepted. Khulood Ali Al Zaabi (2016) [3] had showed the possible man-in- the-middle attack on LinkedIn applications and artefacts left by the application in the host device. But the applications are becoming secure day by day. And also, if the host device is not available for examination or the scenario of investigating the potential suspects without spooking them, we need a procedure to find the artefacts that are being left those suspects activities. For that, the analysis of network traffic of those instant messaging applications is important. Daniel Walnycky et al. (2015) [4] showed the deep analysis of both the device storage and net- work artefacts of 20 popular instant messaging applications. But they were focused on the network artefacts that were available unencrypted. They discarded the encrypted traffic. M.A.K. Sudozai et al. (2017) explored the secure traffic of instant messaging application Viber [5]. Viber is one of the applications which is providing end-to- end encryption. They identified the ports which were used by the Viber servers and separated the Viber traffic from the whole traffic. They used payload sizes to determine the user activities and voice/video calls. Another one of the most popular instant messaging application is WhatsApp. It has more than

1 billion users across the globe. This wide scale of user base shows the importance of the artefacts that can be extracted regarding this application. F. Karpisek et al. (2015) [6] decrypted the WhatsApp encrypted traffic using pidgin plugin. They decrypted many valuable information like WhatsApp phone numbers, phone call metadata like duration, timestamp and WhatsApp voice codec. Fu- Ching Tsai et al. (2018) [7] explored the network artefacts of WhatsApp traces. They found the end parties of a WhatsApp call using Wire- shark. They proposed the voice calls follow STUN protocol and filtered and analyzed those packets to find end parties IP addresses. Instead of analyzing the packets and their patterns, Machine learning techniques were also used to extract information about those secure apps. Zhenlong Yuan et al. (2014) [8] explored the net- work artefacts of skype using sequence signatures of the packets and infer information using ma- chine learning techniques. Scott E. Coull and Kevin P. Dyer (2014) [9] analyzed the encrypted iMessage service and used machine learning techniques to infer information about the underlying operating systems and the language used. Nikunj Malik et al (2017) [10] explored the ICMP pings to the smartphones and used the inter packet space of the traffic to profile the operating system of the device. M.A.K.Sudozai et al. (2018) [11] proposed a framework for profiling the secure apps from the encrypted traffic. They proposed to identify the ports used by the servers and their IP ranges using a firewall and analyze the behavior of the network traffic for various user activities. It was proposed as a common framework and can be applied to any secure instant messaging applications.

3 Research Methodology

This section citates about the various tools which are used to helps to analysis the result and discussed about the methodology used to obtain the results. Line is one of the messaging applications which serves many facilities to the user such as texting, voice call, video call ,emojis and sharing the contact information and location accuracy. This application also provides end to end encryption in the aspects of data privacy of the users. In addition, an overview of LINE's encryption protocols is explained in this paper. According to this survey, LINE is using a specific method called Letter Sealing method to encrypt the messages and VoIP services. These services enable end to end protocol. LINE works on ECDH (Elliptic Curve Diffie - Hellman) for over curve 25519 for key exchange. Encryption is done using Advanced Encryption Standard (AES) -256 in CBC mode and hashing the values are done by SHA-256 algorithm for hashed values. For the LINE's messaging application VoIP encryption protocol, the curve uses are secp256rl. The main purpose of this is to convert RTP to SRTP sessions [20]. LINE provides text messages, video/voice call, location accuracy, file sharing and much more features added on to this application. These provided end to end services to the user. In order to identify the user activities using the application can be identified by following the TCP and UDP flows, byte patterns, and payload sizes of the pattern. For analyzing the large amount of network traffic, samples were collected. Wireshark is used for analyzing the network traffic. Device used were Lenovo idea-pad 33OS which runs with windows 10 with the processor Intel i5-8250U. The devices and the Line version used in the study was discussed below.



For this study, the environmental lab setup is shown in the **Figure 1**. Wire-shark is used to analyze the network traffic. To monitor the network traffic we use monitoring device as Lenovo idea-pad 33OS in which it has an internet access. The mobile hotspot in the monitoring device is connected to the target mobile devices. The target mobile devices are installed with the LINE messaging application. The background process is blocked. The user activities are profiled, and the network traffic analysis is done by wire-shark is examined for the study.

4 Design Specification

This area includes the broad forensic examination done on the secure LINE traffic on the different platforms including android, windows, mac OS, iOS gadgets. The analysis is based on the signature like byte patterns and payload sizes of the packets in the network traffic. The specific signatures were found based on the corresponding user activities like voice/video calling, text messages, file sharing, emojis, contact information, profile information and location based accuracy.

These analyses can be done on various platforms and the findings are evaluated.

4.1 Three-way handshake protocol:

A three-way handshake protocol is a method used in a TCP/IP network to establish the connection between a client and server to exchange SYN and ACK (acknowledgement) packets before the communication of actual data starts.

iOS ip address: 192.168.137.228 Android ip address: 192.168.137.97

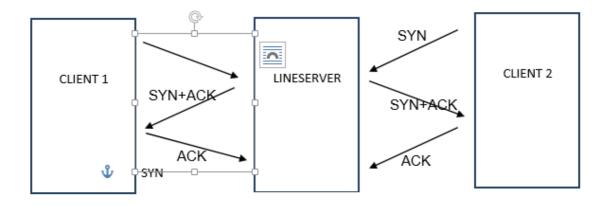


Figure 2 Mapping with Line Server using three-way handshake protocol

Platform	OS version	Device model	LINE version	
Android	9.1.0		8.10.2	
		OnePlus 5,Note 9		
iOS	13.1	iPhone xr,8 plus	8.8.0	
Windows	10	lenovo ideapad 330S	5.8.0	Fig

ure 3 The devices used for analysis

In Figure 3, this details about the devices which are used to analysis the network traffic

In **figure 2** depicted above explains when client initiates a communication, a pairing is done with the line server using the three-way handshake protocol. Once the destination client receives the communication sent by client 1, client 2 creates a connection with line server using the three-way handshake protocol again.

As Instant Messaging apps are becoming widespread, it is essential to find the forensic artefacts that are being left by the usage of those secure apps [16]. In this paper, the behavior of LINE secure traffic is extensively studied and signatures for various LINE user activities were observed based on the patterns in the traffic. A large number of simulations were done to derive conclusions about the user activities signatures. The accuracy and the reliability of the traffic profiling was increased by analyzing in different platforms with different devices in various networks [18]. Through necessary screenshots and tables, detection of LINE secure

traffic and their user activities classification into voice/video calls, text messaging and file sharing is demonstrated in this paper.

5 Implementation

5.1 Android and IOS:

5.1.1 ATTACHMENTS:

There are different types of attachments that can be sent using LINE application. They are images, videos, other file types like pdf, txt, docx , html etc., contacts, and locations. There are also other user activities like sending emoji stickers, viewing profile information etc. All observed byte patterns and their related events are given below.

			T 🖢 🚍 🔍 Q	Q III	
Apply	a display filter <0	Ctrl-/>			Expressi
	Time	Source	Destination	Protocol	Lengt Info
	1 0.000000	192.168.137.97	192.168.137.1	DNS	90 Standard query 0xc123 A android.prod.cloud.netflix.com
	2 0.014208	192.168.137.1	192.168.137.97	DNS	209 Standard query response 0xc123 A android.prod.cloud.netflix.com CNAME prod.cloud.geo.netflix.com CNAME prod
	3 1.531706	192.168.137.97	74.125.193.95	TCP	74 37508 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=17086093 TSecr=0 WS=256
	4 1.549488	74.125.193.95	192.168.137.97	TCP	74 443 → 37508 [SYN, ACK] Seq=0 Ack=1 Win=62392 Len=0 MSS=1430 SACK_PERM=1 TSval=130405403 TSecr=17086093 WS=250
	5 1.551602	192.168.137.97	74.125.193.95	ICMP	102 Destination unreachable (Port unreachable)
	6 1.849068				74 [TCP Retransmission] 443 → 37508 [SYN, ACK] Seq=0 Ack=1 Win=62392 Len=0 MSS=1430 SACK_PERM=1 TSval=130405703.
	7 1.920961	192.168.137.97	74.125.193.95	ICMP	102 Destination unreachable (Port unreachable)
	8 3.119289	192.168.137.97	224.0.0.251	MDNS	103 Standard query 0x0004 PTR _C1EB68AEsubgooglecasttcp.local, "QM" question PTR _googlecasttcp.local, "
	9 3.915764	74.125.193.95	192.168.137.97	TCP	74 [TCP Retransmission] 443 → 37508 [SYN, ACK] Seq=0 Ack=1 Win=62392 Len=0 MSS=1430 SACK_PERM=1 TSval=130407703
1	0 3.971167	192.168.137.97	74.125.193.95	ICMP	102 Destination unreachable (Port unreachable)
1	1 4.573618	192.168.137.97	125.209.252.17	TLSv1.2	650 Application Data, Application Data, Application Data, Application Data, Application Data
1	2 4.614759	125.209.252.17	192.168.137.97	TCP	66 443 → 40052 [ACK] Seq=1 Ack=585 Win=8068 Len=0 TSval=4258874100 TSecr=17086396
1	3 4.863335	125.209.252.17	192.168.137.97	TLSv1.2	159 Application Data
1	4 4.865933	125.209.252.17	192.168.137.97	TLSv1.2	207 Application Data
1	5 4.869023	125.209.252.17	192.168.137.97	TLSv1.2	147 Application Data
1	6 4.890831	192.168.137.97	125.209.252.17	TCP	66 40052 → 443 [ACK] Seq=585 Ack=94 Win=347 Len=0 TSval=17086430 TSecr=4258874349
1	7 4.891186	192.168.137.97	125.209.252.17	TCP	66 40052 → 443 [ACK] Seq=585 Ack=235 Win=351 Len=0 TSval=17086430 TSecr=4258874352
1	.8 4.891187	192.168.137.97	125.209.252.17	TCP	66 40052 → 443 [ACK] Seq=585 Ack=316 Win=351 Len=0 TSval=17086430 TSecr=4258874355
1	9 4.899570	192.168.137.97	125.209.252.14	TLSv1.2	577 Application Data
2	0 4.899904	192.168.137.97	125.209.252.14	TLSv1.2	97 Encrypted Alert
2	1 4.900851	192.168.137.97	125.209.252.14	TCP	74 38648 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=17086431 TSecr=0 WS=256
2	2 4.941489	125.209.252.14	192.168.137.97	TCP	54 443 → 38522 [RST] Seq=1 Win=0 Len=0
2	3 4 941798	125 209 252 14	192 168 137 97	TCP	54 443 → 38522 [RST] Seg=1 Win=0 Len=0

> Internet Protocol Version 4, Src: 192.168.137.97, Dst: 192.168.137.1 > User Datagram Protocol, Src Port: 34972, Dst Port: 53

Figure 4 Analysis of network traffic while sending attachments

This **Figure 4** explains about the network traffic analyzed while sending the attachment file between two devices using Line application.

5.1.2 EMOJI

Emoji are the smileys used in instant messages and web pages. These are more like emoticons.

A	pply a display filter \ldots <	Ctrl-/>			Expres
No.	Time	Source	Destination	Protocol L	engt Info
	1 0.000000	74.125.193.188	192.168.137.97	TCP	66 5228 → 39545 [ACK] Seq=1 Ack=1 Win=253 Len=0 TSval=1180786202 TSecr=16979947
	2 2.132186	192.168.137.97	125.209.252.17	TLSv1.2	278 Application Data, Application Data, Application Data, Application Data
	3 2.175242	125.209.252.17	192.168.137.97	TCP	66 443 → 39206 [ACK] Seq=201 Ack=213 Win=9489 Len=0 TSval=4257446115 TSecr=16980160
	4 2.177545	192.168.137.97	125.209.252.17	TLSv1.2	137 Application Data
	5 2.227741	125.209.252.17	192.168.137.97	TCP	66 443 → 39206 [ACK] Seq=201 Ack=284 Win=9489 Len=0 TSval=4257446168 TSecr=16980166
	6 2.230309	192.168.137.97	125.209.252.17	TLSv1.2	110 Application Data
	7 2.267614	125.209.252.17	192.168.137.97	TCP	66 443 → 39206 [ACK] Seq=201 Ack=328 Win=9489 Len=0 TSval=4257446206 TSecr=16980172
	8 2.777061	192.168.137.97	224.0.0.251	MDNS	119 Standard query 0x0001 PTR _C1EB68AEsubgooglecasttcp.local, "QU" question PTR _CA5E8412subgoogleca
	9 3.864614	192.168.137.97	224.0.0.251	MDNS	119 Standard query 0x0002 PTR _C1EB68AEsubgooglecasttcp.local, "QM" question PTR _CA5E8412subgoogleca
	10 4.881420	192.168.137.97	224.0.0.251	MDNS	119 Standard query 0x0003 PTR _C1EB68AEsub. googlecasttcp.local, "QM" question PTR _CA5E8412subgoogleca
	11 7.081019	203.104.142.91	192.168.137.228	TLSv1.2	97 Encrypted Alert
	12 7.081386	203.104.142.91	192.168.137.228	TCP	66 443 → 63454 [FIN, ACK] Seq=32 Ack=1 Win=269 Len=0 TSval=31471802 TSecr=1196582728
	13 7.124092	192.168.137.228	203.104.142.91	TCP	66 63454 → 443 [ACK] Seq=1 Ack=32 Win=1025 Len=0 TSval=1196604129 TSecr=31471802
	14 7.124094	192.168.137.228	203.104.142.91	TCP	66 63454 → 443 [ACK] Seq=1 Ack=33 Win=1025 Len=0 TSval=1196604129 TSecr=31471802
	15 7.125311	192.168.137.228	203.104.142.91	TLSv1.2	97 Encrypted Alert
	16 7.126356	192.168.137.228	203.104.142.91	TCP	66 63454 → 443 [FIN, ACK] Seq=32 Ack=33 Win=1025 Len=0 TSval=1196604131 TSecr=31471802
	17 7.378730	203.104.142.91	192.168.137.228	TCP	54 443 → 63454 [RST] Seq=33 Win=0 Len=0
	18 7.379028	203.104.142.91	192.168.137.228		54 443 → 63454 [RST] Scg-33 Win-0 Len-0
	19 7.841316	192.168.137.97	74.125.193.103	TLSv1.2	627 Application Data
	20 7.859499	74.125.193.103	192.168.137.97	TCP	66 443 → 43426 [ACK] Seq=1 Ack=562 Win=337 Len=0 TSval=2765424226 TSecr=16980731
	21 7.862101	192.168.137.97	74.125.193.103	TLSv1.2	100 Application Data
	22 7.876565	74.125.193.103	192.168.137.97	TCP	66 443 → 43426 [ACK] Seq=1 Ack=596 Win=337 Len=0 TSval=2765424244 TSecr=16980735
	23 7 901320	74 125 193 103	192 168 137 97	TLSv1 2	655 Application Data

Figure 5 Traffic analysis while sending emoji

This Figure 5 shows the traffic analyzed while sending the emoji using Line application.

5.1.3 CALLING (VIDEO CALL AND VOICE CALL)

One of main difference between the voice and video calls in the LINE app traffic was the packet sizes. Among the hundreds of samples traffics, it was constantly observed that voice call packets will be between 50 bytes to 250 bytes. Packets of video calling will be between 800 bytes to 1200 bytes. When changing from video to voice call or vice versa, the change in packet sizes were also observed clearly.

olv a display filter <		<u> </u>	• unlater	Expr
Time	Source	Destination	Protocol	Lenat Info
1 0.000000	192,168,137,228	224.0.0.251	MDNS	181 Standard guery 0x0000 PTR companion-link. tcp.local, "OM" guestion PTR homekit. tcp.local, "OM" guestion
2 0.000277	fe80::63:22fc:feaa:		MDNS	201 Standard guery 0x0000 PTR companion-link. tcp.local, "QM" guestion PTR homekit. tcp.local, "QM" guestion
3 1,218202	1.0.0.1	192,168,137,97	TCP	54 853 → 43130 [FIN, ACK] Seg=1 Ack=1 Win=31 Len=0
4 1.218434	1.0.0.1	192.168.137.97	тср	54 [TCP Out-Of-Order] 853 → 43130 [FIN. ACK] Seg=1 Ack=1 Win=31 Len=0
5 1,255579	192,168,137,97	1.0.0.1	TCP	66 43130 → 853 [ACK] Seg=1 Ack=4294967266 Win=388 Len=0 SLE=1 SRE=2
6 1.255915	192.168.137.97	1.0.0.1	TCP	74 [TCP Dup ACK 5#1] 43130 → 853 [ACK] Seg=1 Ack=4294967266 Win=388 Len=0 SLE=1 SRE=2 SLE=1 SRE=2
7 1.262143	1.0.0.1	192,168,137,97		85 [TCP Retransmission] 853 - 43130 [FIN, PSH, ACK] Seq=4294967266 Ack=1 Win=31 Len=31
8 1.265111	192.168.137.97	1.0.0.1	TLSv1.2	85 Encrypted Alert
9 1.265436	192.168.137.97	1.0.0.1	TCP	54 43130 → 853 [FIN, ACK] Seq=32 Ack=2 Win=388 Len=0
10 1.282256	1.0.0.1	192.168.137.97	TCP	54 853 → 43130 [RST] Seg=2 Win=0 Len=0
11 1.283206	1.0.0.1			54 853 → 43130 [RST] Seq=2 Win=0 Len=0
12 1.906536	192.168.137.228	192.168.137.1	DNS	84 Standard query 0xc556 A 9-courier.push.apple.com
13 1.930130	192.168.137.1	192.168.137.228	DNS	203 Standard guery response 0xc556 A 9-courier.push.apple.com CNAME 9.courier-push-apple.com.akadns.net CNAME
14 1.935408	192.168.137.228	17.57.146.84	TCP	78 52073 → 5223 [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1194560021 TSecr=0 SACK PERM=1
15 1.949168	17.57.146.84	192.168.137.228	TCP	74 5223 → 52073 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK PERM=1 TSval=4230862455 TSecr=11945
16 1.953226	192.168.137.228	17.57.146.84	TCP	66 52073 → 5223 [ACK] Seg=1 Ack=1 Win=131712 Len=0 TSval=1194560038 TSecr=4230862455
17 1.953553	192.168.137.228	17.57.146.84	TLSv1.2	288 Client Hello
18 1.970638	17.57.146.84	192.168.137.228	TCP	66 5223 → 52073 [ACK] Seq=1 Ack=223 Win=30080 Len=0 TSval=4230862475 TSecr=1194560038
19 1.972116	17.57.146.84	192.168.137.228	TLSv1.2	1514 Server Hello
20 1.973086	17.57.146.84	192.168.137.228	TCP	1514 5223 → 52073 [ACK] Seq=1449 Ack=223 Win=30080 Len=1448 TSval=4230862476 TSecr=1194560038 [TCP segment of a
21 1.973297	17.57.146.84	192.168.137.228	TLSv1.2	594 Certificate, Server Key Exchange, Server Hello Done
22 1.975822	192.168.137.228	17.57.146.84	TCP	66 52073 → 5223 [ACK] Seq=223 Ack=2897 Win=128832 Len=0 TSval=1194560061 TSecr=4230862476
23 1 975823	192 168 137 228	17 57 146 84	TCP	66 52073 - 5223 [ACK] Sen=223 Ack=3425 Win=128320 Len=0 TSval=1194560061 TSecr=4230862476

Figure 6 network traffic analysis during video and voice call

This **Figure 6** shows the network traffic while doing voice call and the packet size are increased if it changes to video call. The video or voice call are analyzed by the packet size.

5.1.4 TEXT

In Text messages, there are two different events. They are user start typing and user sending the message. For both these events, two different signatures were observed during the behavior analysis.

ply a display filter	actual (se			
Time	Source	Destination	Protocol	Lenot Info
1 0.000000	74,125,193,119	192,168,137,97		2 122 Application Data
2 0.036259	192.168.137.97	74.125.193.119	ICMP	150 Destination unreachable (Port unreachable)
3 1,502060	192.168.137.97	125,209,252,17		2 334 Application Data
4 1.545147	125.209.252.17	192.168.137.97	TCP	66 443 + 39078 [ACK] Seq=1 Ack=169 Win=8312 Len=0 TSval=120842379 TSecr=16929795
5 1.545461	125.209.252.17	192.168.137.97	TLSv1.2	
6 1.586573	192.168.137.97	125.209.252.17	TCP	66 39078 + 443 [ACK] Seq=169 Ack=78 Win=347 Len=0 TSval=16929806 TSecr=120842380
7 1.803221	125.209.252.17	192.168.137.97		137 Application Data
8 1.880409	192,168,137,97	125,209,252,17	TCP	66 39078 + 443 [ACK] Seq=169 Ack=149 Win=347 Len=0 TSval=16929835 TSecr=120842637
9 2.059574	125.209.252.17	192.168.137.97	TLSv1.2	
10 2.062424	192.168.137.97	125,209,252,17	TCP	66 39078 + 443 [ACK] Seq=169 Ack=374 Win=351 Len=0 TSval=16929853 TSecr=120842893
11 2.064062	125.209.252.18	192.168.137.228	TLSv1.2	
12 2.120973	192.168.137.97	125,209,252,17	TLSV1.2	
13 2.176003	192.168.137.228	125,209,252,18	TCP	66 52102 + 443 [ACK] Seq=1 Ack=254 Win=2044 Len=0 TSval=1195445138 TSecr=4239298515
14 2,197584	125,209,252,17	192.168.137.97	TCP	66 443 + 39078 [ACK] Seq=374 Ack=240 Win=8312 Len=0 TSval=120843033 TSecr=16929859
15 2.213062	192,168,137,228	125,209,252,18	TLSv1.2	
16 2.219174	192.168.137.228	125.209.252.18		2 155 Application Data
17 2.251805	125,209,252,18	192,168,137,228	TCP	66 443 + 52102 [ACK] Seq=254 Ack=157 Win=8553 Len=0 TSval=4239298703 TSecr=1195445174
18 2.262639	125.209.252.18	192.168.137.228	TCP	66 443 → 52102 [ACK] Seq=254 Ack=246 Win=8553 Len=0 TSval=4239298715 TSecr=1195445180
19 2.392678	209.85.202.157	192.168.137.97	TLSv1.2	
20 2.395817	192.168.137.97	209.85.202.157	ICMP	150 Destination unreachable (Port unreachable)
21 2.498135	125.209.252.18	192.168.137.228	TLSv1.2	2 152 Application Data
22 2.517564	125.209.252.18	192.168.137.228		188 Application Data
23 2 585425	192 168 137 228	125 209 252 18	TCP	66 52102 → 443 [ACK] Seg=246 Ack=340 Win=2046 Len=0 TSval=1195445543 TSecr=4239298951

Figure 7 Network traffic analysis while sending text messages

The **Figure 7** shows the traffic between two devices while sending the text messages. The signature are analyzed while sending the text.

6 Evaluation

6.1 Android and IOS :

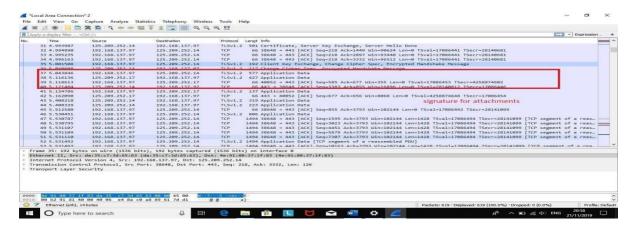
As presented during the implementation part, both android and iOS applications have same signatures for various user activities. Those signatures are:

Event	Payload size	From	То
	(bytes)		
User starts	234	Client	Server
typing	66	Server	Client
User sends the	137	Server	Client
message	66	Client	Server

Table 1 signature of user activities

End of	147	Server	Client
voice/video call	66	Client	Server
Sending	577	Client	Server
	66	Server	Client
video, other file types, contacts)	427	Server	Client
	66	Client	Server
	1.62		C
Start typing	162	Client	Server
emoji stickers	66	Server	Client
Sending Emoji	137	Client	Server
sticker	66	Server	Client

In the above **Table 1** these signatures are table as examination is conducted on hundreds of network samples on both android and iOS devices.



6.1.1 Signature for Attachments:

Figure 8 Signature for attachments

This screenshot, **Figure 8** illustrates about the signature found while sending attachments from two devices. IP address (iOS): **192.168.137.228** and (Android): **192.168.137.97**. .192.168.137.97 send the attachment to 192.168.137.228. The signature found for sending attachments is **577 is from client to server66 ACK message from server to client 427 is from server to client 66 ACK message from client to server.**

6.1.2. Signature for Emoji:

for start
and the second second
oji
googlecas
-
googlecas
Boogrecas
1
goog1

Figure 9 Signature for emoji

This above **Figure 9** shows about the signature found while sending the emoji from two devices. IP address (iOS): **192.168.137.228** and (Android): **192.168.137.97**. 192.168.137.97 send the emoji to

192.168.137.228. The signature found for sending emoji is **162** is from client to server .Client starts typing emoji 66 ACK message from server to client

137 is from client to server. when the client hits the send button. 66 ACK message from server to client

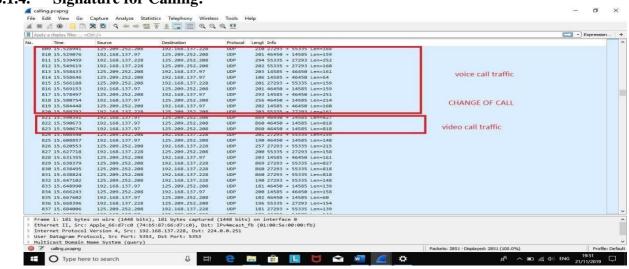
6.1.3. Signature for Texting :

	<ctrl-></ctrl->			Expression.
Time	Source	Destination	Protocol Lengt Info	
1 0.000000	74.125.193.119	192.168.137.97	TLSv1.2 122 Application Data	
2 0.036259	192.168.137.97	74.125.193.119	ICMP 150 Destination unreachable (Port unreachable)	-
3 1.502060	192.168.137.97	125.209.252.17	TLSv1.2 234 Application Data	
4 1.545147	125.209.252.17	192.168.137.97	TCP 66 443 → 39078 [ACK] Seq=1 Ack=169 Win=8312 Len=0 TSval=120842379 TSecr=16929795	
5 1.545461	125.209.252.17	192.168.137.97	TLSv1.2 143 Application Data	
6 1.586573	192.168.137.97	125.209.252.17	TCP 66 39078 → 443 [ACK] Seq=169 Ack=78 Win=347 Len=0 TSval=16929806 TSecr=120842380	
7 1.803221	125.209.252.17	192.168.137.97	TLSv1.2 137 Application Data	
8 1.880409	192.168.137.97	125.209.252.17	TCP 66 39078 → 443 [ACK] Seq=169 Ack=149 Win=347 Len=0 TSval=16929835 TSecr=12084263	7
9 2.059574	125.209.252.17	192.168.137.97	TLSv1.2 291 Application Data	
10 2.062424	192.168.137.97	125.209.252.17	TCP 66 39078 → 443 [ACK] Seq=169 Ack=374 Win=351 Len=0 TSval=16929853 TSecr=12084289	3
11 2.064062	125.209.252.18	192.168.137.228	TLSv1.2 319 Application Data	
12 2.120973	192.168.137.97	125.209.252.17	TLSv1.2 137 Application Data	
13 2.176003	192.168.137.228	125.209.252.18	TCP 66 52102 → 443 [ACK] Seq=1 Ack=254 Win=2044 Len=0 TSval=1195445138 TSecr=4239298	515
14 2.197304	123.209.232.17	192.100.137.97	00 445 4 59010 [RCK] 304-514 RCK-240 R10-0512 [C0140 15081-120045055 13601-109230.	signature exacts
15 2.213062	192.168.137.228	125.209.252.18	TLSv1.2 222 Application Data	0
16 2.219174	192.168.137.228	125.209.252.18	TLSv1.2 155 Application Data	when user
17 2.251805	125.209.252.18	192.168.137.228	TCP 66 443 → 52102 [ACK] Seq=254 Ack=157 Win=8553 Len=0 TSval=4239298703 TSecr=11954	
18 2.262639	125.209.252.18	192.168.137.228	TCP 66 443 → 52102 [ACK] Seq=254 Ack=246 Win=8553 Len=0 TSval=4239298715 TSecr=11954	45180 Series the text
19 2.392678	209.85.202.157	192.168.137.97	TLSv1.2 122 Application Data	
20 2.395817	192.168.137.97	209.85.202.157	ICMP 150 Destination unreachable (Port unreachable)	
21 2.498135	125.209.252.18	192.168.137.228	TLSv1,2 152 Application Data	
22 2.517564	125.209.252.18	192.168.137.228	TLSv1.2 188 Application Data	
23 2 585425	192 168 137 228	125 209 252 18	TCP 66 52102 - 443 [ACK] Sen=246 Ack=340 Win=2046 Len=0 TSval=1195445543 TSecn=42392	38951
			1144 bits) on interface 0 : da:35:c7:3d:65:63 (da:35:c7:3d:65:63)	
	ol Version 4, Src: 125.		8. Seg: 1. Ack: 169. Len: 77	
ansport Layer		rt: 443, Dst Port: 390	s, seq: 1, Ack: 169, Len: //	
	security			

Figure 10 Signature for text messages

This screenshot, Figure 10 details about the signature found while sending text from two devices. IP address (iOS):192.168.137.228 and (Android): 192.168.137.97 192.168.137.97 send the text to 192.168.137.228. The signature found for sending the text messages is 234 is from client to server. When the client starts typing the text 66 ACK message from server to

client .137 is from server to client. When the sender hits the send button 66 ACK message from client to server.



6.1.4. Signature for Calling:

Figure 11 Signature for video and voice call

In Figure 11 explains about the signature found for the video and voice call traffic. This can be analyzed by packet size. For filtering the voice and video call traffic, the packets size can be used. Filter(Voice call)- frame.len ≤ 250 && frame.len ≥ 50 Filter(Video call)- frame.len ≤ 1200 && frame.len ≥ 800

Windows

Windows is one of the most popular and common operating system used across the world. LINE has provided a windows desktop application .The various user activities and their signatures are given below.

6.1.5 Voice and video call

For voice & video calling feature, the signature doesn't change from the android and IOS traffic [15]. The voice call data payload stays between 50-250 bytes and video call packets stay between 800-1200 bytes in size. When changing from video to voice call or vice versa, the change in packet size were also observed clearly.

Other User Activities:

Unlike android/iOS applications, the windows LINE application doesn't have different signature for different user activities. Because, for both sending and receiving, the activities

like text messages, attachment files and emoji's have same signature [19]. Hence, profiling those activities will be harder than the android/iOS traffic. But they can be classified based on the position of the signature chunk in that session. There are also few different signatures for activities like profile viewing. The various signatures according to the windows LINE application are:

Event (send/receive)	Payload size (bytes)	Position	From	То
text message	110	At the end of the	Client	Server Client
	56	session	Server Client	Server
	116		Server	Client
	56			
Attachments	110	Beginning of the	Client	Server Client
(audio/video	56	session	Server Client	Server
files)	116		Server	Client
	56			
Attachments	110	End of the session	Client	Server Client
(other file types,	56		Server Client	Server
contacts)	116		Server	Client
	56			

Table 2. Signatures of Windows LINE app.

From the above **Table 2 signatures of windows LINE app**, it is clear that profiling the user activities of windows LINE application will be tricky, since most of the user activities share the same signature [14] with few differences. Hence it is important to note those few differences like position of the signature chunk to successfully profile those activities.

6.3 IOS to IOS6.3.1. ATTACHMENTS

6 3,4474354 102.11 7 3.4804471 125.24 8 3,484633 192.11 9 3,484634 192.11 10 3.526589 125.24 11 3,529876 125.24 12 3,530428 125.24 13 3,530458 125.24	Destination 58.137.1 192.168.137.46 181.137.46 192.168.137.46 192.22.13 192.168.137.46 18.137.46 125.299.252.13 19.252.13 192.168.137.46 19.252.13 192.168.137.46 19.252.13 192.168.137.46 19.252.33 192.168.137.46 19.252.33 192.168.137.46	Protoci Lengt Info DHS 151 Standard query response 0xbf60 A obs-de-line-apps.com CNAME obs-de-addr.line-apps.com A 125.209.252.13 A 125 DHS 151 Standard query response 0xbf60 A obs-de-line-apps.com CNAME obs-de-addr.line-apps.com A 125.209.252.13 A 125 NEW 70 02311034 fom ungeschable (cons ungeschable) TCP 74 A43 + 62017 [SVM, ACK] Seq=1 Ack=1 kin=14300 Lene0 MSs-1440 SACK PEIM=1 Tsval=11220993 TSecr=1506073833 k5=5 TCP 66 62017 - 443 [AcK] Seq=1 Ack=1 kin=131328 Lene0 TSval=1506073875 TSecr=31220993 TLSv1.2 581 Clinet He1l0 TCP 66 43 - 62017 [AcK] Seq=1 Ack=18 kin=15872 Lene0 TSval=31221040 TSecr=1506073875 TLSv1.2 514 Secree He1l0
6 3,4474354 102.11 7 3.4804471 125.24 8 3,484633 192.11 9 3,484634 192.11 10 3.526589 125.24 11 3,529876 125.24 12 3,530428 125.24 13 3,530458 125.24	58,137,46 192,168,137,1 192,213 192,168,137,46 188,137,46 125,269,252,13 192,252,13 192,168,137,46 192,252,13 192,168,137,46	IC#9 70 Destination unreachable (Port unreachable) TCP 77 4431 cost[sav1 [SW1 Ack] Benet Ack-1 kin:14480 Ennet MS5-1448 SACK PERM-1 Tsval=31220993 Tsecr=150807383 k5=5_ TCP 66 2037 + 443 [Ack] Secr1 Ack-1 kin:131328 Ennet Tsval=106073875 Tsecr=31220993 TCP 66 443 + 62317 [Ack] Secr1 Ack-518 kin:15572 Ennet Tsval=32040 Tsecr=1508073875
7 3.480471 125.24 8 3.484633 192.11 9 3.484634 192.11 10 3.526589 125.24 11 3.529876 125.24 12 3.530208 125.24 13 3.530458 125.24	99.252.13 192.168.137,46 18.137.46 125.209.252.13 18.137.46 125.209.252.13 19.252.13 192.168.137.46 19.252.13 192.168.137.46	TCP 74 443 + 62017 [SWN, ACK] seq=4 Ack+1 kin-14400 Lene MS-1440 SACK PIRMA I Sval-3122093 Tsecr=1500973833 kS=5. TCP 66 62017 + 443 [ACK] Seq=1 Ack+1 kin-131328 Lene M Stval=1500973875 TSecr=31220993 Tsecr=1500973833 kS=5. TCP 66 62017 + 443 [ACK] Seq=1 Ack+1 kin-131328 Lene M Stval=31097375 TSecr=31220993 Tsecr=1500973875 TCP 66 443 + 62017 [ACK] Seq=1 Ack+518 kin=15872 Lene M Stval=31221040 TSecr=1500973875
8 3.484633 192.10 9 3.484634 192.10 10 3.526589 125.20 11 3.526876 125.20 12 3.530208 125.20 13 3.530458 125.20	58.137.46 125.209.252.13 58.137.46 125.209.252.13 99.252.13 192.168.137.46 99.252.13 192.168.137.46	TCP 66 62017 + 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=1506973675 TSecr=31220993 TL5V1.2 583 Client Hello TCP 66 443 + 62017 [ACK] Seq=1 Ack=518 Win=15872 Len=0 TSval=31221040 TSecr=1506973875
9 3.484634 192.10 10 3.526589 125.20 11 3.529876 125.20 12 3.530208 125.20 13 3.530458 125.20	58.137.46 125.209.252.13 59.252.13 192.168.137.46 59.252.13 192.168.137.46	TLSV1.2 583 Client Hello TCP 66 443 → 62017 [ACK] Seq=1 Ack=518 Win=15872 Len=0 TSval=31221040 TSecr=1506973875
10 3.526589 125.20 11 3.529876 125.20 12 3.530208 125.20 13 3.530458 125.20	9.252.13 192.168.137.46 99.252.13 192.168.137.46	TCP 66 443 → 62017 [ACK] Seq=1 Ack=518 Win=15872 Len=0 TSval=31221040 TSecr=1506973875
11 3.529876 125.20 12 3.530208 125.20 13 3.530458 125.20	99.252.13 192.168.137.46	
12 3.530208 125.20 13 3.530458 125.20		TLSv1.2 1514 Server Hello
13 3.530458 125.20	9.252.13 192.168.137.46	
		TCP 1514 443 → 62017 [ACK] Seq=1449 Ack=518 Win=15872 Len=1448 TSval=31221042 TSecr=1506973875 [TCP segment of a reas…
	9.252.13 192.168.137.46	TLSv1.2 501 Certificate, Server Key Exchange, Server Hello Done
	58.137.46 125.209.252.13	TCP 66 62017 → 443 [ACK] Seq=518 Ack=2897 Win=128384 Len=0 TSval=1506973924 TSecr=31221042
	58.137.46 125.209.252.13	TCP 66 62017 → 443 [ACK] Seq=518 Ack=3332 Win=128000 Len=0 TSval=1506973924 TSecr=31221042
	125.209.252.13 125	TCP 66 [TCP Window Update] 62017 + 443 [ACK] Seq=518 Ack=3332 Win=131072 Len=0 TSval=1506973924 TSecr=31221042
	58.137.46 125.209.252.13	TLSv1.2 192 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
	99.252.13 192.168.137.46	TLSv1.2 117 Change Cipher Spec, Encrypted Handshake Message
	125.209.252.13 125.209	TCP 66 62017 → 443 [ACK] Seq=644 Ack=3383 Win=130944 Len=0 TSval=1506973972 TSecr=31221093
	58.137.46 125.209.252.13	TLSv1.2 1087 Application Data
	58.137.46 125.209.252.13	TCP 1494 62017 → 443 [ACK] Seq=1665 Ack=3383 Win=131072 Len=1428 TSval=1506973979 TSecr=31221093 [TCP segment of a re…
	58,137,46 125,209,252,13	TCP 1494 62017 → 443 [ACK] Seq=3093 Ack=3383 Win=131072 Len=1428 TSval=1506973979 TSecr=31221093 [TCP segment of a re…
	9.252.13 192.168.137.46	TCP 66 443 → 62017 [ACK] Seq=3383 Ack=3093 Win=20480 Len=0 T5val=31221149 T5ecr=1506973978
	58,137,46 125,209,252,13	TCP 1494 62017 → 443 [ACK] Seq=4521 Ack=3383 Win=131072 Len=1428 TSval=1506974025 TSecr=31221149 [TCP segment of a re…
	58.137.46 125.209.252.13	TCP 1494 62017 → 443 [ACK] Seq=5949 Ack=3383 Win=131072 Len=1428 TSval=1506974025 TSecr=31221149 [TCP segment of a re_ TCP 1494 62017 → 443 [ACK] Seq=5747 Ack=3383 Win=131072 Len=1428 TSval=1506974025 TSecr=31221149 [TCP segment of a re_
	58.137.46 125.209.252.13 18.137.46 125.209.252.13	TCP 1494 62017 + 443 [ACK] Seq-7377 ACK=3303 Win=131072 Len=1428 TSval=1506074025 TSecr=31221149 [TCP segment of a re- TCP 1404 62017 + 443 [ACK] Seq-RR05 ACK=33103 Win=131072 Len=1428 TSval=1506074025 TSecr=31221140 [TCP segment of a re-
thernet II, Src: 4e:91:80 Iternet Protocol Version	4, Src: 192.168.137.1, Dst: 239 c Port: 57519, Dst Port: 1900	Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)

Figure 12 signature for attachment in IOS

This above **Figure 12** shows about the signature found while sending the attachment from two devices. IP address (iOS): **192.168.137.228** and (iOS): **192.168.137.46** 192.168.137.46 send the attachment to 192.168.137.228. The signature found for sending attachments is **583** is from client to server . **66** ACK message from server to client **501** is from server to client. **66** ACK message from client to server.

6.3.3 TEXT

me . 321402					Expression.	
221402	Source	Destination	Protocol	Lengt Info		
	fe80::63:22fc:feaa:	ff02::fb	MDNS	201 Standard query 0x0000 PTR _companion-link. tcp.local, "QM" question PTR _homekit. tcp.local,	"OM" question PTR	
.073062	fe80::63:22fc:feaa:	ff02::16	ICMPv6	90 Multicast Listener Report Message v2		-
0.331087	192.168.137.228	224.0.0.251	MDNS	181 Standard guery 0x0000 PTR companion-link. tcp.local, "OM" guestion PTR homekit. tcp.local,	"OM" guestion PTR	-
0.331431	fe80::63:22fc:feaa:	ff02::fb	MDNS	201 Standard guery 0x0000 PTR companion-link, tcp.local, "OM" guestion PTR homekit, tcp.local,	"OM" question PTR	
7.690504	192.168.137.46	203.104.153.129	SSLv2	230 Encrypted Data, Continuation Data		-
7,953434	203.104.153.129	192,168,137,46	SSLv2	155 Encrypted Data, Continuation Data		
7.959712	203.104.153.129	192.168.137.46	SSLv2	136 Encrypted Data, Continuation Data		
8.021333	192,168,137,46	203.104.153.129	TCP	66 62000 → 443 [ACK] Seg=343 Ack=276 Win=2046 Len=0 TSval=1504958442 TSecr=4155466710		
8.021334	192.168.137.46	203.104.153.129	TCP	66 62000 → 443 [ACK] Seq=343 Ack=346 Win=2045 Len=0 TSval=1504958442 TSecr=4155466716		
0.459016	192.168.137.46	203,104,153,129	SSLv2	231 Encrypted Data, Continuation Data		
0.712939	203.104.153.129	192.168.137.46	SSLv2	155 Encrypted Data, Continuation Data		
0.716748	203.104.153.129	192.168.137.46	SSLv2	136 Encrypted Data, Continuation Data		
0.786292	192.168.137.46	203.104.153.129	TCP	66 62000 + 443 [ACK] Seq=508 Ack=435 Win=2046 Len=0 TSval=1504961204 TSecr=4155469469		
0.786294	192.168.137.46	203.104.153.129	TCP	66 62000 → 443 [ACK] Seq=508 Ack=505 Win=2045 Len=0 TSval=1504961204 TSecr=4155469473		
0.806002	192.168.137.228	192.168.137.1	DHCP	342 DHCP Request - Transaction ID 0xee2dd0e6		
0.817335	192.168.137.1	192.168.137.228	DHCP	344 DHCP ACK - Transaction ID 0xee2dd0e6		
0.953876	192.168.137.228	224.0.0.251	MDNS	181 Standard query 0x0000 PTR companion-link, tcp.local, "OU" question PTR homekit, tcp.local,	"OU" question PTR	002
0.954239	fe80::63:22fc:feaa:	ff02::fb	MDNS	201 Standard guery 0x0000 PTR companion-link. tcp.local, "QU" guestion PTR homekit. tcp.local,	"QU" question PTR	
1.934695	192.168.137.228	224.0.0.2	IGMPV2	46 Leave Group 224.0.0.251		-
1.934854	192.168.137.228	224.0.0.251	IGMPv2	46 Membership Report group 224.0.0.251		
1.958104	192.168.137.228	224.0.0.251	MDNS	181 Standard guery 0x0000 PTR companion-link, tcp.local, "OM" guestion PTR homekit, tcp.local,	"OM" question PTR	
1.958513	fe80::63:22fc:feaa:	ff02::fb	MDNS	201 Standard guery 0x0000 PTR companion-link. tcp.local, "OM" guestion PTR homekit. tcp.local,	"QM" question PTR	-
3 028127	192 168 137 228	2 18 225 5	TI 51/1 2	90 Application Data		-
	a. 331431 .690504 .953434 .959712 8.021333 8.021333 8.021334 8.021334 9.712399 8.716748 9.786294 9.786294 9.786294 9.886092 9.887335 9.954239 1.934854 1.958184 1.958513 4.028127	3.33.41 fe00:103:22fc:fean: 2.069040 102:106:137.46 7.95344 203.104:153.129 9.05712 203.104:153.129 3.02134 102.106:137.46 3.02134 102.106:137.46 3.02135 102.106:137.46 3.02136 102.106:137.46 3.71209 203.104:153.129 3.716748 203.104:153.129 3.706294 102.106:137.46 3.706294 102.106:137.46 3.060292 102.106:137.46 3.060292 102.106:137.46 3.053876 102.106:137.42 3.053876 102.106:137.21 3.053876 102.106:137.21 3.054294 102.106:137.72 3.054295 102.106:137.72 3.054295 102.106:137.72 3.054295 102.106:137.72 3.054295 102.106:137.728 3.054295 102.106:137.728 3.054295 102.106:137.728 3.054295 102.106:137.728 3.054295 102.106:137.728 3.05335	3.33.41 feB0:163122fc;feas1_f02:rfb 7.00504 103122fc;feas1_f02:rfb 7.00504 103:126137.46 203:104:133.129 7.05334 203:104:135.129 102:166.137.46 7.05334 203:104:135.129 102:166.137.46 8.02133 192:168.137.46 203:104:153.129 7.02399 201:161.137.46 203:104:153.129 7.72399 201:161.137.46 203:164.137.74 7.72399 201:104:153.129 102:164.137.74 7.726204 192:166.137.46 203:104:153.129 7.726204 192:166.137.46 203:164.153.129 8.06002 192:166.137.46 203:164.153.129 9.051266 137.228 224.66.137.1 9.052367 102:166.137.228 224.66.251 9.054230 109:166.137.228 224.60.251 9.054231 109:166.137.228 224.60.251 9.054231 109:166.137.228 224.60.251 9.051261 107:161.137.238 246.00.251 1.09:161.137.228 224.60.251 1.09:161.137.228 246.00.251	3.33.31 fel0::63:22fc:feas. ff02:1fb PDMS 7.055343 203:164:173.129 122.164.137.46 253.104.153.129 554.123 7.955343 203:104.153.129 122.166.137.46 554.123 7.955343 203:104.153.129 122.166.137.46 554.123 8.02133 122.166.137.46 203.104.153.129 TCP 8.02133 122.166.137.46 203.104.153.129 TCP 8.02133 122.166.137.46 203.104.153.129 TCP 8.71234 203.104.153.129 TCP 203.104.153.129 TCP 8.712349 203.104.153.129 122.166.137.46 254.104.153.129 TCP 8.716478 203.104.153.129 122.166.137.46 203.104.153.129 TCP 8.706020 192.166.137.228 124.66.357.228 TCP 203.104.153.129 TCP 8.05022 192.166.137.228 224.66.357.228 TCP 204.055.176 MDMS 9.053267 102.166.137.228 224.66.251 MDMS MDMS 1099.124 9.054237 102.166.137.228 <th>3.33341 fe80:i0122fc:feasff02:rb MOMS 20. Standard query 0:00000 PTR _companion-link_tcp.local, "Q" question PTR _homekit_tcp.local, "Q," question PTR _homekit_tcp.local, 1,991011111111111111111111111111111111</th> <th>3,333.11 fel00:i012/fcfreatff02:rb POMS 201 Standard query 0x0000 PTR _companion-link_tcp.local, "QM" question PTR _homekit_tcp.local, "QM" ques</th>	3.33341 fe80:i0122fc:feasff02:rb MOMS 20. Standard query 0:00000 PTR _companion-link_tcp.local, "Q" question PTR _homekit_tcp.local, "Q," question PTR _homekit_tcp.local, 1,991011111111111111111111111111111111	3,333.11 fel00:i012/fcfreatff02:rb POMS 201 Standard query 0x0000 PTR _companion-link_tcp.local, "QM" question PTR _homekit_tcp.local, "QM" ques

Figure 14 signature for text in IOS

This above **Figure 14** shows about the signature found while sending the text from two devices. IP address (iOS): **192.168.137.228** and (iOS): **192.168.137.46** 192.168.137.46 send the text to 192.168.137.228. The signature found for sending text messages is **231 is from client to server**.

66 ACK message from server to client 136 is from server to client.

66 ACK message from client to server.

6.3.4 VIDEO AND VOICE CALLING

Source 125.209.252.208 192.168.137.97	Destination 192.168.137.228	Protocol	ngt Info	
	192.168.137.228	LIDE		
103 168 137 07			10 27293 → 55335 Len=168	
	125,209,252,208	UDP	01 46450 → 14585 Len=159	
192.168.137.228	125,209,252,208	UDP	94 55335 → 27293 Len=252	
192.168.137.228	125.209.252.208	UDP	02 55335 → 27293 Len=160	
	192.168.137.97	UDP	03 14585 → 46450 Len=161	
125.209.252.208	192.168.137.97	UDP		
125,209,252,208	192,168,137,228	UDP		
192.168.137.97	125,209,252,208	UDP	01 46450 → 14585 Len=159	
125,209,252,208	192,168,137,97	UDP		
192.168.137.228	125.209.252.208	UDP	03 55335 → 27293 Len=161	
192.168.137.97	125.209.252.208	UDP	60 46450 → 14585 Len=818	
125.209.252.208	192,168,137,228	UDP	01 27293 → 55335 Len=159	
192,168,137,228	125,209,252,208	UDP		
125.209.252.208	192.168.137.97	UDP	03 14585 → 46450 Len=161	
	192.168.137.228 125.290.252.208 125.290.252.208 125.290.252.208 192.168.137.97 125.290.252.208 192.168.137.97 125.290.252.208 192.168.137.228 192.168.137.97 192.168.137.97 192.168.137.97 192.168.137.97 192.168.137.97 192.202.252.208	192.166.137.226 125.209.252.208 155.209.252.208 152.166.137.07 155.209.252.208 152.166.137.07 155.209.252.208 152.166.137.07 155.209.252.208 152.166.137.07 155.209.252.208 152.166.137.07 155.209.252.208 152.166.137.07 155.209.252.208 152.166.137.07 155.209.252.208 152.166.137.07 155.209.252.208 152.166.137.07 155.209.252.208 152.166.137.07 155.209.252.208 152.166.137.27 155.209.252.208 152.166.137.28 155.209.252.208 152.166.137.28 155.209.252.208 152.166.137.28 155.209.252.208 152.166.137.28 155.209.252.208 152.166.137.28 155.209.252.208 152.166.137.28 155.209.252.208 152.166.137.28 155.209.252.208 152.166.137.28 155.209.252.208 152.166.137.28 155.209.252.208 152.166.137.28 155.209.252.208 152.166.137.28 155.209.252.208 152.166.137.28 155.20	192.168.137.228 125.299.252.208 UOP 2 135.299.252.298 135.168.137.97 UOP 1 135.299.252.298 132.168.137.97 UOP 1 135.299.252.298 132.168.137.97 UOP 1 135.299.252.298 132.168.137.97 UOP 1 135.299.252.298 132.168.137.97 UOP 2 135.299.252.298 132.168.137.97 UOP 2 135.299.252.298 132.168.137.97 UOP 2 135.299.252.298 UOP 2 1 132.168.137.97 125.299.252.268 UOP 2 132.168.137.97 125.299.252.288 UOP 1 135.299.252.298 104.137.282 UOP 1 135.299.252.298 132.168.137.97 125.299.252.298 UOP 1 135.299.252.298 132.168.137.27 UOP 2 1 135.299.252.298 132.168.137.28 UOP 1 1 135.299.252.298 132.168.137.28 UOP 1 1<	192.104.137.22 125.209.252.200 107.00 202.5335 27230 107.106 155.209.252.200 152.106.137.77 107 007 203.5455 46456 1em-164 155.209.252.201 152.106.137.77 007 108.14567 46456 1em-164 155.209.252.201 152.106.137.77 007 203.1455 46456 1em-164 125.106.137.77 125.209.252.208 007 203.1455 46456 1em-231 125.106.137.77 125.209.252.208 007 203.1455 46456 1em-231 125.106.137.77 125.209.252.208 007 203.1455 46456 1em-231 126.106.137.201 125.209.252.208 007 803.4558 47203 1em-161 125.106.137.707 125.209.252.208 007 804.4549 1455 1em-161 125.209.252.208 126.137.207 127.209.252.208 007 804.4549 1455 1em-161 125.209.252.208 126.137.277 007 127.209.455.358 1em-151 1em-251 125.209.252.208 126.137.278 007 803.1455 1em-151

Figure 15 signature for calling in IOS

In Figure 15, Signature of video and voice call is determined by the packet size.

For filtering the voice and video call traffic, the packets size can be used. Filter(Voice call)- frame.len <= 250 && frame.len >= 50 Filter(Video call)- frame.len <= 1200 && frame.len >= 800

7 Conclusion and Future Work

The above analysis implies the profiling the user activities even on the encrypted network traffic. The analysis explains detail about the various signature profiling the user activities like voice/ video calling, sending text messages, attachments, file sharing and location accuracy so on. The findings of the signatures are based on the payload sizes and the frequency of the packets sent and receive. The network traffic analysis can be done in different platforms such as an Android, iOS, windows and macOS. The mobile devices such as android and iOS have a numerous sensitive information regarding the user activities. These analyses will help the investigators or any security professionals in their research. The analysis is done with single user communications, there is an enhanced feature such as group chat messaging which needs a clear study for the future works. Further research can be done with group messaging and group calling features of the LINE applications can be performed on various platforms. To conclude from the above analysis, it is clear analysis of profiling the user activities based on the signatures that found in the traces of analyzing the network traffic. This information gathered are sensitive and can be useful for the person in the Law Enforcement Agency or Corporate investigation cases. The suspect user activities can be found without spooking the user. If a decryption key for the encrypted LINE traffic is analyses in the future, these signatures are useful for decrypting the data .

References

- 1. F. A. Awan, Forensic examination of social networking applications on smartphones,, Vol. 2015.
- F. N. Dezfouli, A. Dehghantanha, Brett, Eterovic- Soric & Kim-Kwang Raymond Choo (2015)Inves- tigating Social Networking applications on smart- phones detecting Facebook, Twitter, LinkedIn and Google+ artefacts on Android and iOS platforms, Australian Journal of Forensic Sciences 48 (4) 469–488.
- 3. K. A. A. Zaabi, Android Forensics: Investigating Social Networking Cybercrimes against Man-in-the- Middle Attacks, 2016.
- 4. D. Walnycky, I. Baggili, A. Marrington, J. Moore, F. Breitinger, Network and device forensic analysis of Android social-messaging applications, Digital In- vestigation, Volume 14 (1), S77-S84, ISSN 1742- 2876.
- 5. M. A. K. Sudozai, N. Habib, S. Saleem, A. A. Khan, Signatures of Viber Security Traffic, Journal of Dig- ital Forensics, Security and Law 12 (2).
- 6. F. Karpisek, I. Baggili, F. Breitinger, WhatsApp net- work forensics: Decrypting and understanding the WhatsApp call signaling messages.
- 7. F. C. Tsai, E. C. C. D. Y. Kao, WhatsApp network forensics: Discovering the communication payloads behind cybercriminals, Korea (South, 2018.
- 8. Z. Yuan, C. Du, X. Chen, D. W. Y. Xue, SkyTracer: Towards fine- grained identification for Skype traffic via sequence signatures, Honolulu, HI, 2014.
- 9. S. E. K. P. Dyer, Traffic Analysis of Encrypted Mes- saging Services: Apple iMessage and Beyond, SIG- COMM Comput. Commun. Rev 44 (2014) 5–11.
- 10.N. Malik, J. Chandramouli, P. Suresh, K. Fairbanks, L. W. W. H. Robinson, Using network traffic to verify mobile device forensic artifacts, NV, 2017.
- 11.M. A. K. S. S. Saleem, Profiling of secure chat and calling apps from encrypted traffic,2018.
- 12. 'LINE Transparency Report' (LINE Corporation)
- 13.N. &. B. I. &. M. A. (. F. a. o. s. n. a. o. m. d. D. I. 9. S. 1. Al Mutawa, Forensic analysis of social networking applications on mobile devices, 2012.
 - 14. T. V. a. R. L. Artur Kane, The data privacy-preserving way, 2018.

- 15. F. &. B. I. &. B. F. (. W. n. f. D. a. u. t. W. c. s. m. D. I. 1. 1. Karpisek, WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages.
- 16. F.Karpisek, WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages, 2015.
- 17. F. A. Awan, Forensic examination of social networking applications on smartphones, 2018.
- 18. J. Fran, cois, Network traffic analysis (for encrypted traffic and security monitoring), 2017.
- 19. P. &. C. M. &. C. P. &. D. M. (. A. s. o. m. f. e. t. c. a. a. I. J. o. N. M. 2. 1. Velan, A survey of methods for encrypted traffic classification and analysis, 2019.
- 20. C. K. Ambreen F.A.H1, Forensic Analysis of Social Applications, 2012.