A study of ten popular Android mobile VoIP applications: Are the communications encrypted?

Abdullah Azfar
Information Assurance
Research Group
University of South Australia
abdullah.azfar@mymail.unisa.edu.au

Kim-Kwang Raymond Choo
Information Assurance
Research Group
University of South Australia
raymond.choo@unisa.edu.au

Lin Liu
School of Information Technology
and Mathematical Sciences
University of South Australia
lin.liu@unisa.edu.au

Abstract

Mobile Voice over Internet Protocol (mVoIP) applications have gained increasing popularity in the last few years, with millions of users communicating using such applications (e.g. Skype). Similar to other forms of Internet and telecommunications, mVoIP communications are vulnerable to both lawful and unauthorized interceptions. Encryption is a common way of ensuring the privacy of mVoIP users. To the best of our knowledge, there has been no academic study to determine whether mVoIP applications provide encrypted communications. In this paper, we examine Skype and nine other popular mVoIP applications for Android mobile devices, and analyze the intercepted communications to determine whether the captured voice and text communications are encrypted (or not). The results indicate that most of the applications encrypt text communications. However, voice communications may not be encrypted in six of the ten applications examined.

1. Introduction

In recent years, Voice over Internet Protocol (VoIP) communication is increasingly used by individuals, businesses and government agencies. A recent study by the Australian Government Department of Broadband, Communications and the Digital Economy [1], for example, found that the number of 'adults using voice over internet protocol (VoIP) rose by nearly 21 per cent in the year to June 2012, to 4.3 million'. It is, perhaps, unsurprising as VoIP provides voice and video communications which are cost effective, and in some cases, free (e.g. Skype to Skype call).

VoIP communications can be intercepted, either lawfully (e.g. by a law enforcement agency authorized by a wiretap warrant) [2] or by unauthorized actors (e.g. compromising a client machine with malware with the intention of intercepting the voice or video communication before it is encrypted by the VoIP application) [3].

There are various ways to intercept VoIP communication. For example, interceptions can take place at the client devices when the communication is being initiated or during the established communication session. Vulnerabilities in the protocols used in VoIP applications such as Session Initiation Protocol (SIP) [4], H.323 [5], Peer-to-Peer SIP (P2PSIP) [6] can also be exploited for interception purposes [7].

Mobile VoIP (mVoIP) applications such as Skype, ICQ, Viber, Google Talk and Tango use either open standard protocols or proprietary protocols.

Although mVoIP applications are relatively new, they are increasingly used by consumers due to the prevalence of mobile devices such as Android devices. However, the security of mVoIP applications appears to be an understudied area. For example, are the communications using mVoIP applications secure (e.g. encrypted)? Understanding whether communications using mVoIP applications are encrypted would facilitate lawful interceptions (e.g. in deciding what interception techniques to use and what resources are required).

The aim of this research is to determine whether communications (both voice and text) using popular mVoIP applications are encrypted (or not). We examine ten mVoIP applications for Android devices. We then analyze the captured text and voice communications using histogram analysis and measuring the entropy of the captured communication sessions.

The rest of the paper is organized as follows: Sections 2 and 3 provide an overview of VoIP interception techniques (that also apply to mVoIP communications), and the ten popular Android mVoIP applications respectively. Our experimental setup is outlined in section 4. Section 5 presents our experiment results, and in Section 6, we discuss our findings. Finally, Section 7 concludes the paper and suggests future work.

978-1-4799-2504-9/14 $31.00 © 2014 IEEE
DOI 10.1109/HICSS.2014.596

Table 1. VoIP Interception Techniques

VoIP Protocol	Interception Method	Comments
SIP	MITM	Any internal attacker [10] or external attacker who knows the IP address of the target SIP phone can initiate the attack [11, 12]
	Call Establishment hijacking	Two MITM attackers manipulate the call establishment. As a result, the caller thinks the callee is busy, but the callee is unaware of the incoming call [13]
	Call Termination hijacking	Prolongs the duration of established calls by hijacking the normal call termination [13]
	Call Forwarding hijacking	Allows an unanswered incoming call to be forwarded to another phone number [13]
P2PSIP	MITM	Needs an interception filter and interception server. This is effective for one-to-one VoIP communication with callee as the interception target [14]
	Malware	Allows interception of outgoing traffic at the source, which would also allow access to all incoming traffic for the target device [15]
	Intercepting at IP-Layer	Useful when target device uses the same network, but is challenging in mobility mode [15]
	Infiltrating the P2P Network	Uses an enrolment server of the operator and infiltrates routing tables [15]
H.323	Wiretap on Gateways method	Can intercept all calls between H.323 network and the PSTN, but not when calls take place within the H.323 network [16]
	Wiretap Routing on Gatekeeper method	Can intercept calls in the H.323 network, but degrades quality of service for intercepted calls [16]
	Fixed Route Wiretap method	Intercepts each and every call, which might lead to network overload [16]
	Promiscuous Wiretap method	Challenges in detecting traffic in high speed networks [16]

2. Overview of VoIP Interception

In VoIP communication, codecs such as G.711, G.729 and Full Rate GSM are used to encode and compress voice signals into digital data, which can complicate the identification of the communication session. Wang, Chen and Jajodia [8] and Chen, Wang and Jajodia [9] proposed watermarking techniques to detect communication between two parties using peer-to-peer (P2P) VoIP.

Verscheure, Vlachos, Anagnostopoulos, Frossard, Bouillet and Yu [17] proposed another method to identify the pair of participating parties in a VoIP communication session. Takahashi and Lee [18] analyzed covert channel interception in VoIP communication. Srivatsa, Iyengar and Liu [19] proposed a flow analysis attack on VoIP networks. In a flow analysis attack, the P2P VoIP traffic is intercepted by exploiting the shortest path nature of the voice flows to identify the communicating parties in a VoIP network. Freire, Ziviani and Salles [20] attempted to detect communications using Skype and Google Talk VoIP with two Goodness-of-Fit tests by measuring the Kolmogorov-Smirnov distance and the $\chi 2$ value. Gomes, Inacio, Pereira, Freire and Monteiro [21] identified the P2P voice communication sessions based on the entropy and properties of different voice codecs. As different protocols and applications exhibit similar characteristics when the same codec is used, they classified the voice data according to the codecs. They examined the length of the packets and entropy to identify the VoIP flows.

Man-in-the-Middle (MITM) attack is another common technique to intercept VoIP communications. Studies have shown that a MITM attack can bypass a VoIP communication, redirect a VoIP call to any third party and manipulate call forwarding options in various services [11]. Such interception is possible as long as the attacker knows the IP address and phone number of the caller [12]. Vrakas, Geneiatakis and Lambrinoudakis [10] demonstrated how an internal user can exploit the SIP REFER method using a MITM attack. Interception techniques involving more than one MITM attacks include Call Hijacking attack. The latter can be categorized into Call Establishment hijacking, Call Termination hijacking and Call Forwarding hijacking [13] (see Table 1). Zhang, Wang, Yang and Jiang [13] explained that Call Establishment hijacking can be achieved by two MITM attackers manipulating the call establishment. The Call Termination hijacking prolongs the duration of the call by manipulating the normal call termination. The Call Forwarding hijacking is performed by forwarding the unanswered call to another number.

P2PSIP allows the user agents (UAs) to connect without going through a dedicated proxy server. Seedorf [15] proposed several interception techniques that can be used in a P2PSIP environment. Examples of the interception techniques are compromising the targeted devices using malware, intercepting the communication at the IP layer, and infiltrating the P2P network. Interception in a P2PSIP environment can be performed by exploiting the basic structure of the Chord algorithm [14] (e.g. use the stabilization method in Chord to acquire the Resource Key of the intercepted target in order to carry out a MITM attack).

Milanovic, Srbljic, Raznjevic, Sladden, Matosevic and Skrobo [16] proposed four methods of intercepting a VoIP communication using the H.323 protocol standard, namely Wiretap on Gateways method, Wiretap Routing on Gatekeeper method, Fixed Route Wiretap method and Promiscuous Wiretap method. Table 1 provides a brief overview of the known interception techniques against various VoIP protocols.

3. mVoIP Applications

There is a range of mVoIP applications available for different mobile devices (e.g. Android, iOS, Windows mobile, Symbian, and Blackberry). Some mVoIP applications support text, voice and video communication, and some support only voice and text communications.

In this section, we examine ten popular Android mVoIP applications that support text, voice and video communications. They are Skype, Google Talk, ICQ, Viber, Nimbuzz, Yahoo, Fring, Vonage, WeChat and Tango. We then install the latest version of the ten applications (at the time of this research – 31 May 2013) on two Android phones (see section 4).

3.1. Overview

Skype uses its own proprietary secure VoIP communication protocol [22]. All the packets of Skype communication are encrypted with the 256-bit Advanced Encryption Standard (AES) [23]. The Skype server certifies the user public keys using either 1536 or 2048-bit RSA certificates [24]. Since it is trivial to determine the target's Skype ID, an attacker can therefore communicate with the target over Skype to determine his/her IP address, even if the target is behind a network address translation (NAT) server [25].

As Skype provides end-to-end encryption, no information about the routing of the data packets could be found from the flow content. The inter-packet timing characteristics are likely to be preserved across intermediate Skype peers and low latency anonymity network. The correlation between anonymous VoIP flows can be determined by the inter-packet timing characteristics. The inter-packet arrival time of VoIP flow is either 20ms or 30ms. The proposed approach of Wang, Chen and Jajodia [8] embeds a unique watermark into the encrypted voice stream by slightly adjusting the timing of selected packets. Sengar, Zhen, Haining, Wijesekera and Jajodia [26] proposed another technique to track VoIP communications over Skype. In this approach, only the callee is known prior to the communication and the caller is tracked during the communication session.

Google Talk uses Extensible Messaging and Presence Protocol (XMPP) [27]. XMPP [28] provides voice communication services through an extension named Jingle [29] . The Jabber stream is not encrypted in Google Talk [30] [31]. Google Talk also uses its own authentication mechanism.

Although Viber is a relatively new inclusion in the mVoIP application community, it has gained popularity among users. Viber does not need a separate login, other than a user name and internet connection – either using WiFi or mobile data network – to send or receive voice calls and messages [32]. Viber provides encrypted text messaging services and scrambles the voice data [33]. Viber uses its own proprietary protocol and the voice packets were not detected in our experiments as RTP streams. Recently, a bug was discovered in Viber where the lock screens of the smart phones can be bypassed to send voice calls and messages [34]. ICQ is another popular mVoIP application, which uses proprietary Open System for Communication in Realtime (OSCAR) messaging protocol. ICQ is available for Windows, Apple iOS, Blackberry and Android platforms, and according to the company's documentation, ICQ does not provide encryption [35].

Yahoo messenger uses its own proprietary protocol to provide instant messaging, photo sharing, PC-to-PC calls, mail alerts, games and other features [36]. A Yahoo voice server was compromised in 2012, which resulted in the theft of 453,491 Yahoo email messages and passwords [37].

Nimbuzz is a communication platform that provides voice and video call services over Internet [38]. Nimbuzz connects with popular instant messaging and social network sites such as Facebook, Google Talk and Yahoo messenger. Nimbuzz uses

Table 2. Supported platforms and Authentication methods of mVoIP Applications

mVoIP Applications	Supported mobile Platforms					Version used in our experiments	Authentication Method
	Android	iOS	Windows	Blackberry	Symbian		
Skype	✓	✓	✓	✓	✓	3.2.0.6673	User name and password
Google Talk	✓	✓	✓	✓	✓	4.2.2-573038	A valid Gmail account
ICQ	✓	✓	✓	✓	✓	4.0.8	User name (a valid Email account) and password
Viber	✓	✓	✓	✓	✓	3.0.1.3	Number of the mobile handset that the application is installed on (e.g. +61 400 123 456)
Nimbuzz	✓	✓	✓	✓	✓	2.4.3	User name and password
Yahoo	✓	✓	✓	✓	✓	1.8.3	A valid Yahoo account
Fring	✓	✓			✓	4.3.0.20	Number of the mobile handset that the application is installed on
Vonage	✓	✓				2.1.1	Number of the mobile handset that the application is installed on
WeChat	✓	✓	✓	✓	✓	4.5.1	Number of the mobile handset that the application is installed on
Tango	✓	✓	✓			2.10.47400	Number of the mobile handset that the application is installed on

XMPP as its primary protocol [39]. Similar to nimbuzz is another communication platform Fring [40]. Fring is a P2P VoIP service provider, which uses Dynamic Video Quality (DVQ) technology for video calls.

Tango [41] uses its own protocol and provides free VoIP calling services between Tango users. Vonage provides a VoIP application – Vonage Mobile [42]. WeChat [43] is another popular VoIP application developed by the Chinese company, Tencent (the same company that developed Tencent QQ, reportedly one of the widest used instant messaging application in China).

Most of the mVoIP applications examined in this paper run on Android, iOS, Windows, Blackberry and Symbian platforms. We used the most recent version as of 31 May 2013 in the experiments. The authentication method varies between mVoIP applications (Table 2).

3.1. Are the communications encrypted?

It is relatively straightforward to determine whether text messages sent using instant messaging (IM) applications are encrypted or not, by analyzing the captured packets. However, determining whether the captured voice communication is encrypted is less straightforward due to a number of reasons:

a) As explained earlier, codecs are used to encode and compress voice signals into digital data. This digitization process scrambles the voice communication, and, consequently, it is hard to determine whether the captured packets are encrypted or not.
b) To decode the captured voice data, we need to use the right decoder. In cases such as open source VoIP applications based on SIP, the payload type of the captured data indicates the codec used in the encoding. However in the case of proprietary VoIP applications (e.g. Skype), there is no indication of the payload type in the captured packets. The captured TCP or UDP communication will only indicate unassigned payload types.

4. Experiments

In our experiments, we used two LG Google Nexus 4 Android phones with Android version 4.2.2. Ten VoIP applications (see Table 2) were examined in our experiments. We used the WiFi network as communication channel.

4.1. Setup

An Android application named Shark for Root was used to capture network traffic in pcap format. The mVoIP applications were run on both phones one at a time. Voice and text messages were captured separately. For each of the ten applications, we captured voice data in both directions for 10 minutes. We played the same song on the phones for all the ten applications in order to maintain consistency. The process was repeated with a different song to get a second set of sample data. This was done to ensure the reliability of the experiment. .

We then sent a series of text messages using the ten applications and captured the communications using Shark for Root to analyze the captured text communications.

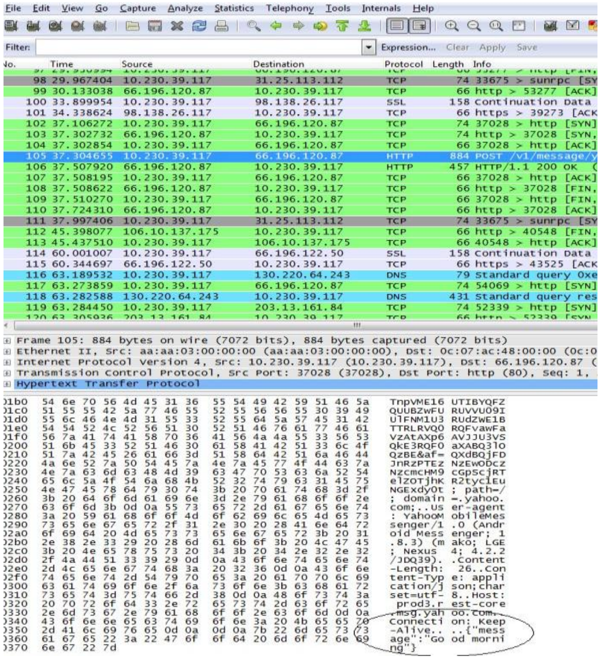
To ensure that there was no other traffic, all other applications that could generate Internet traffic were turned off. As we are interested only in voice and text data, all other packets were filtered out and only the UDP packets containing the voice and text data were analyzed.

4.2. Analysis of captured voice packets

We applied the following two statistical methods to find out whether the packets are encrypted or not.

The first method is based on the frequency distribution of the byte values. We used the PcapHistogram tool [44] to read the payload of captured packets and plot a histogram with frequency on the Y axis and the byte values (in hex) on the X axis. One of the three possible conclusions can be drawn from the histogram:

a) If there is a cluster of byte values around the region 0x41 to 0x5A (representing English uppercase alphabet set) and 0x61 to 0x7A (representing English lowercase alphabet set) in the histogram with other regions barely covered, then this indicates the packets contain plaintexts and the captured session is not encrypted. For VoIP data, as they are encoded, it is very unlikely to get a cluster in these two regions.
b) If the byte values are evenly distributed without any clustered region, then the session is most probably encrypted.
c) These packets might be obfuscated using a XOR key if the frequency distributions are clustered around a region that does not represent the English characters. In this case, the captured session is most probably not encrypted. VoIP data are likely to follow this pattern due to the encoding mechanism if they are not encrypted



```
No.   Time       Source           Destination      Protocol Length Info
  98 29.967404  10.230.39.117    31.25.113.112    TCP        74 33675 > sunrpc [SY
  99 30.133038  66.196.120.87    10.230.39.117    TCP        66 http > 53277 [ACK]
 100 33.899954  10.230.39.117    98.138.26.117    SSL       158 Continuation Data
 101 34.338624  98.138.26.117    10.230.39.117    TCP        66 https > 39273 [ACK
 102 37.106272  10.230.39.117    66.196.120.87    TCP        74 37028 > http [SYN]
 103 37.302732  66.196.120.87    10.230.39.117    TCP        74 http > 37028 [SYN,
 104 37.302854  10.230.39.117    66.196.120.87    TCP        66 37028 > http [ACK]
 105 37.304655  10.230.39.117    66.196.120.87    HTTP      884 POST /v1/message/y
 106 37.507920  66.196.120.87    10.230.39.117    HTTP      457 HTTP/1.1 200 OK  (
 107 37.508195  10.230.39.117    66.196.120.87    TCP        66 37028 > http [ACK]
 108 37.508622  66.196.120.87    10.230.39.117    TCP        66 http > 37028 [FIN,
 109 37.510270  10.230.39.117    66.196.120.87    TCP        66 37028 > http [FIN,
 110 37.724310  66.196.120.87    10.230.39.117    TCP        66 http > 37028 [ACK]
 111 37.997406  10.230.39.117    31.25.113.112    TCP        74 33675 > sunrpc [SY
 112 45.398077  106.10.137.175   10.230.39.117    TCP        66 http > 40548 [FIN,
 113 45.437510  10.230.39.117    106.10.137.175   TCP        66 40548 > http [ACK]
 114 60.001007  10.230.39.117    66.196.122.50    SSL       158 Continuation Data
 115 60.344697  66.196.122.50    10.230.39.117    TCP        66 https > 43525 [ACK
 116 63.189532  10.230.39.117    130.220.64.243   DNS        79 Standard query 0xe
 117 63.273859  10.230.39.117    66.196.120.87    TCP        74 54069 > http [SYN]
 118 63.282588  130.220.64.243   10.230.39.117    DNS       431 Standard query res
 119 63.284450  10.230.39.117    203.13.161.84    TCP        74 52339 > http [SYN]
 120 63.305936  203.13.161.84    10.230.39.117    TCP        66 http > 52339 [SYN

⊞ Frame 105: 884 bytes on wire (7072 bits), 884 bytes captured (7072 bits)
⊞ Ethernet II, Src: aa:aa:03:00:00:00 (aa:aa:03:00:00:00), Dst: 0c:07:ac:48:00:00 (0c:0
⊞ Internet Protocol Version 4, Src: 10.230.39.117 (10.230.39.117), Dst: 66.196.120.87 (
⊞ Transmission Control Protocol, Src Port: 37028 (37028), Dst Port: http (80), Seq: 1,
⊞ Hypertext Transfer Protocol

01b0  54 6e 70 56 4d 45 31 36  55 54 49 42 59 51 46 5a   TnpVME16 UTIBYQFZ
01c0  51 55 55 42 5a 77 46 55  52 55 56 56 55 30 39 49   QUUBZwFU RUVVU09I
01d0  55 6c 46 4e 4d 31 55 33  52 55 64 5a 57 45 31 42   UlFNM1U3 RUdZWE1B
01e0  54 54 52 4c 52 56 51 30  52 51 46 76 61 77 46 61   TTRLRVQ0 RQFvawFa
01f0  56 7a 41 74 41 58 70 36  41 56 4a 4a 55 33 56 53   VzAtAXp6 AVJJU3VS
0200  51 6b 45 33 52 51 46 30  61 58 41 42 51 33 6c 4f   QkE3RQF0 aXABQ3lO
0210  51 7a 42 45 26 61 66 3d  51 58 64 42 51 6a 46 44   QzBE&af= QXdBQjFD
0220  4a 6e 52 7a 50 54 45 7a  4e 7a 45 77 4f 44 63 7a   JnRzPTEz NzEwODcz
0230  4e 7a 63 6d 63 48 4d 39  63 47 70 53 63 6a 52 54   NzcmcHM9 cGpScjRT
0240  65 6c 5a 4f 54 6a 68 4b  52 32 74 79 63 31 45 75   elZOTjhK R2tyc1Eu
0250  4e 47 45 78 64 79 30 74  3b 20 70 61 74 68 3d 2f   NGExdy0t ; path=/
0260  3b 20 64 6f 6d 61 69 6e  3d 2e 79 61 68 6f 6f 2e   ; domain =.yahoo.
0270  63 6f 6d 3b 0d 0a 55 73  65 72 2d 61 67 65 6e 74   com;..Us er-agent
0280  3a 20 59 61 68 6f 6f 4d  6f 62 69 6c 65 4d 65 73   : YahooM obileMes
0290  73 65 6e 67 65 72 2f 31  2e 30 20 28 41 6e 64 72   senger/1 .0 (Andr
02a0  6f 69 64 20 4d 65 73 73  65 6e 67 65 72 3b 20 31   oid Mess enger; 1
02b0  2e 38 2e 33 29 20 28 6d  61 6b 6f 3b 20 4c 47 45   .8.3) (m ako; LGE
02c0  3b 20 4e 65 78 75 73 20  34 3b 20 34 2e 32 2e 32   ; Nexus  4; 4.2.2
02d0  2f 4a 44 51 33 39 29 0d  0a 43 6f 6e 74 65 6e 74   /JDQ39). .Content
02e0  2d 4c 65 6e 67 74 68 3a  20 32 36 0d 0a 43 6f 6e   -Length:  26..Con
02f0  74 65 6e 74 2d 54 79 70  65 3a 20 61 70 70 6c 69   tent-Typ e: appli
0300  63 61 74 69 6f 6e 2f 6a  73 6f 6e 3b 63 68 61 72   cation/j son;char
0310  73 65 74 3d 75 74 66 2d  38 0d 0a 48 6f 73 74 3a   set=utf- 8..Host:
0320  20 70 72 6f 64 33 2e 72  65 73 74 2d 63 6f 72 65    prod3.r est-core
0330  2e 6d 73 67 2e 79 61 68  6f 6f 2e 63 6f 6d 0d 0a   .msg.yah oo.com..
0340  43 6f 6e 6e 65 63 74 69  6f 6e 3a 20 4b 65 65 70   Connecti on: Keep
0350  2d 41 6c 69 76 65 0d 0a  0d 0a 7b 22 6d 65 73 73   -Alive.. ..{"mess
0360  61 67 65 22 3a 22 47 6f  6f 64 20 6d 6f 72 6e 69   age":"Go od morni
0370  6e 67 22 7d                                        ng"}
```

Figure 1. Plaintext in Yahoo messenger

The second method is to calculate the entropy of the payloads of the captured packets by using Shannon entropy, which measures the uncertainty associated with a random variable [45]. Given a random variable X with $\boldsymbol{N}$ possible values $\{x_1, x_2, \ldots., x_n\}$, its entropy can be calculated as:

$$H(X) = -\sum_{i=0}^{N-1} p_i \log_2 p_i \ ,$$

where $p_i = P(X = x_i)$

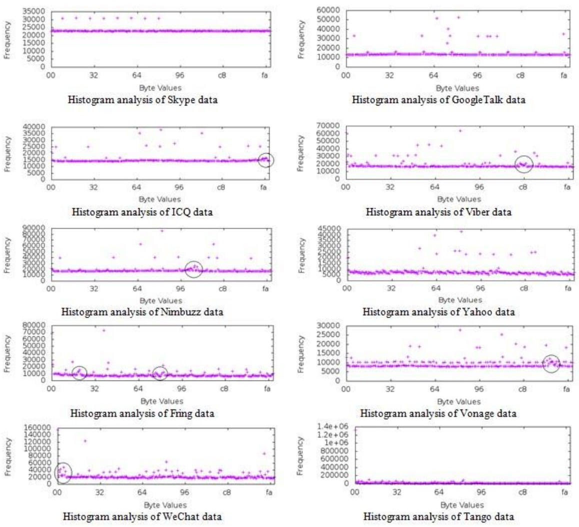
Then the minimum average number of bits per character is:

$$numCharacters = Upper\ bound\ of\ H(X)$$

English language has a very low entropy of 2.3 bits per character on average due to its predictable nature. However for encrypted packets, the bits are more widespread (evenly distributed) and the entropy value becomes higher (greater than 5 bits per character on average).

As stated earlier, voice data is encoded and due to the encoding mechanism, the entropy becomes higher than the English language entropy even if no encryption is applied to the voice data. However, the lack of encryption results in frequent changes in the entropy with high and low peaks, while encryption mechanism distributes the characters evenly. Therefore the entropy of encrypted voice data is high and the entropy distribution is even (i.e. no sudden changes of high or low peaks). This can be used as the indicator in identifying encrypted voice data.

In our analysis, the entropy of the captured packets is calculated by the Shannon's entropy measurement tool named pyNetEntropy [46].


Histogram analysis of Skype data

Histogram analysis of GoogleTalk data

Histogram analysis of ICQ data

Histogram analysis of Viber data

Histogram analysis of Nimbuzz data

Histogram analysis of Yahoo data

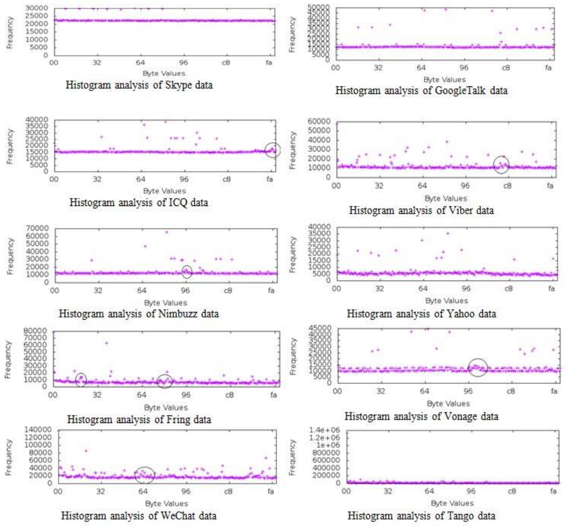
Histogram analysis of Fring data

Histogram analysis of Vonage data

Histogram analysis of WeChat data

Histogram analysis of Tango data

Figure 2. Histogram analysis (Sample 1)


Histogram analysis of Skype data

Histogram analysis of GoogleTalk data

Histogram analysis of ICQ data

Histogram analysis of Viber data

Histogram analysis of Nimbuzz data

Histogram analysis of Yahoo data

Histogram analysis of Fring data

Histogram analysis of Vonage data

Histogram analysis of WeChat data

Histogram analysis of Tango data

Figure 3. Histogram analysis (Sample 2)

5. Findings

Three tests were performed after capturing the mVoIP communication sessions. In the first experiment, we analyzed the captured text messages to look for a plaintext. The second and third experiments were performed to determine whether voice communications using the ten mVoIP applications were encrypted.

5.1. Text data analysis

After analyzing the pcap files containing instant messages with Wireshark, we found that the mVoIP applications provide encrypted or secure communication, with the exception of Yahoo messenger. No plaintext data was visible from the captured Skype conversation packets. Google Talk IMs were also not visible and no clues were found from the captured packets. The protocol name was also not visible from Google Talk data.

Analyzing the captured session for ICQ revealed that ICQ uses JavaScript Object Notation (JSON) for text messages. The text messages were encoded using JSON's encryption mechanism. However, the profiles of the communicating parties were visible in plaintext in the captured packets. The profile information included the user ID, first name, last name, gender, relationship status, job description, and religion.

We were also able to determine that both Nimbuzz and Tango use XMPP to encrypt the text communications, and both Tango and Vonage use Transport Layer Security (TLS) and HTTPS to provide secure communication.

For Yahoo, it was interesting to note that the captured text messages sent by the user of the device where the messages were captured were in plaintext. However, the captured text messages received by the user were encrypted. A snapshot of the plaintext message of Yahoo messenger is shown in Figure 1. The plaintext "Good morning" is marked with a circle.

5.2. Voice data analysis using Histogram

Our findings of the captured voice data by analyzing the pcap files with pcap histogram are shown in Figures 2 and 3. The histogram for Skype was consistent in both samples. There is no cluster in any region of the histograms, which suggests that Skype VoIP may be encrypted.

For Google Talk, the histogram analysis showed frequency distribution of bytes with no clusters for both samples. This suggests that Google Talk may also be encrypted.

For ICQ, the results were interesting. A small cluster was shown in the region 0xFA in both samples. The clusters are marked with circles. This indicates that ICQ voice data is not encrypted, but due to the encoding of voice data, the histogram has a cluster in the 0xFA region.

The histogram of the sessions captured for Viber showed clusters in the region 0xC8 (see Figure 2) and in the region 0xC2 (see Figure 3). Other regions had scattered byte distribution. The clustered regions reflect the scrambling mechanism used by Viber. The analysis of Nimbuzz data also revealed clusters in the regions 0x99 (see Figure 2) and 0x96 (see Figure 3).

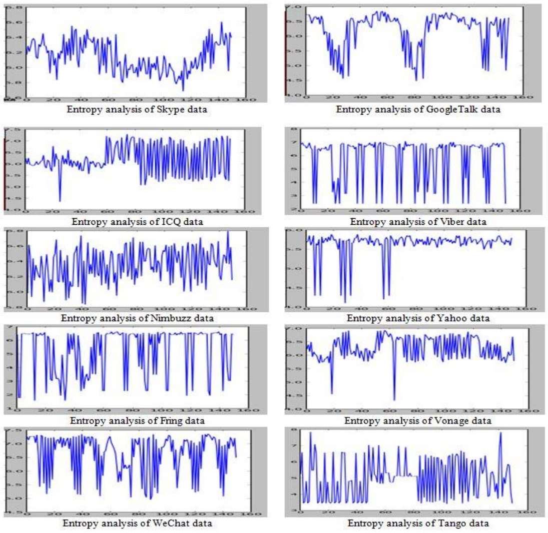


Figure 4. Entropy analysis (Sample 1)

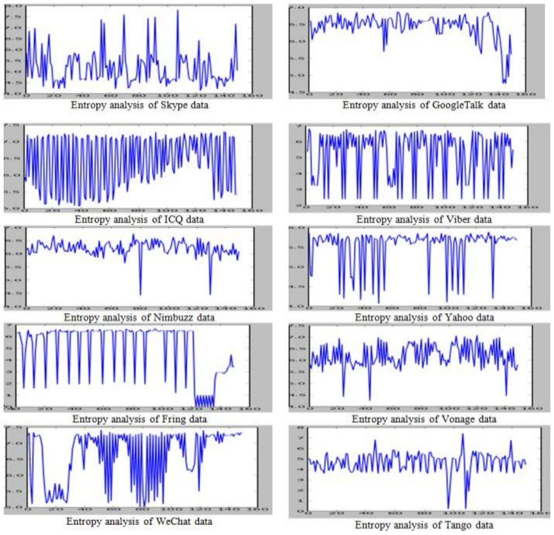


Figure 5. Entropy analysis (Sample 2)

The results from the analysis of the captured Yahoo voice data showed even distribution of the bytes with no cluster in any region. On the other hand, analysis of Fring data indicated that there were clusters in the 0x20 and 0x80 regions in both samples. This suggests that voice communication in Fring is not encrypted.

The Vonage data analysis showed a cluster in the region 0xE1 (see Figure 2) and a cluster in the region 0x9F (see Figure 3). Other regions had scattered distribution of bytes. The analysis of WeChat data showed a cluster in both samples in region 0x03 (see Figure 2) and 0x64 (see figure 3). The presence of these clusters in Vonage and WeChat voice data indicates the absence of encryption.

Tango voice data histogram analysis revealed no clustering in any region, which suggests Tango may use encryption mechanism.

5.3. Entropy analysis

Due to the encoding of voice data, the entropy of the captured packets is expected to be higher than the entropy of English language. The experiment results showed a relatively high entropy value for all the applications. But for encrypted data, the entropy value must be constant with only a minor fluctuation. A high fluctuation in the entropy indicates the bits are not randomly distributed in the captured files.

The entropy analysis of the first sample of Skype data produced a result of 5.7 bits per character to 6.6 bits per character in Figure 4.The change in entropy was even and the value varied within 1.0 bit per character as shown in Figure 4. In the second sample in Figure 5, the variation in entropy was higher.

For Google Talk, the entropy results were between 4.6 and 6.6 bits per character. The fluctuation was higher than Skype. As shown in Figure 4, the fluctuation occurred slowly. There was no sudden spike in the entropy of Google Talk. This gradual change of entropy can be due to the use of encryption. In Figure 5, the entropy was consistent around the region of 6.0 and 7.0 bits per character. However, there is a sudden drift towards the end, which can be classified as outlier due to network noise.

Results from ICQ were again interesting. The results were very consistent. As shown in Figure 4, it is in the range of 6.0 and 6.2 bits per character during the beginning, but there is a sudden spike where the entropy reduced to 4.5 bits per character. Then again the entropy increased to 6.0 bits per character. After a while, a continuous fluctuation began within the range of 5.2 to 7.3 bits per character. In other words, there is an uneven distribution of entropy. As shown in Figure 5, the entropy distribution is constantly changing. The uneven distribution of entropy suggests the absence of encryption in ICQ voice data.

Viber also produced an uneven entropy distribution within the range of 2.5 to 7.0 bits per character - see Figures 4 and 5. The fluctuation was very high and the entropy change was continuous. There was hardly any region where the (Viber) entropy remained constant. This is an indication that Viber communication is scrambled rather than encrypted.

The entropy analysis results of Nimbuzz had a steady distribution with sudden spikes. The average distribution was between 6.1 and 6.5 bits per character. But there were spikes in the range of 5.8 and 6.8 bits per character. In Figure 5, the entropy distribution was very consistent around the region of 6.0 and 6.5 bits per character with two sudden drifts. Overall, the entropy distribution was very even.

Yahoo voice entropy distribution had high spikes

Table 3. Summary findings from the experiments

VoIP Apps	Text communication encrypted? (Yes/No)	Cluster in Histogram Analysis		Entropy Analysis		Voice communication encrypted? (Yes/No)
		Sample1	Sample2	Sample1	Sample 2	
Skype	Yes	No	No	Steady	Steady with sudden changes	Yes
Google Talk	Yes	No	No	Gradual change	Gradual change	Yes
ICQ	Yes	Yes	Yes	Uneven	Steady changes	**No**
Viber	Yes	Yes	Yes	High fluctuation	High fluctuation	**No**
Nimbuzz	Yes	Yes	Yes	Steady changes	Steady changes	Yes
Yahoo	**No (messages sent by user)** Yes (messages received by user)	No	No	High fluctuations in the beginning	High fluctuation	**No**
Fring	Yes	Yes	Yes	High fluctuation	High fluctuation	**No**
Vonage	Yes	Yes	Yes	Steady with few spikes	Steady with few spikes	**No**
WeChat	Yes	Yes	Yes	Even and uneven	Even and uneven	**No**
Tango	Yes	No	No	High fluctuation	Steady changes	Yes

during the beginning of the analysis with values ranging from 4.1 to 5.8 bits per character (see Figure 4). It subsequently remained steady within the range of 5.6 and 5.8 bits per character. As shown in Figure 5, the entropy was at times steady, and at times fluctuating.

The entropy analysis of Fring produced highly varying entropy between 1.5 and 6.5 bits per characters throughout the analysis for both samples.

The overall entropy distributions for Vonage were around the range of 5.7 and 6.9 bits per character in Figure 4, and 5.5 and 6.5 bits per character in Figure 5.

For WeChat, the entropy results varied in the range of 5.0 and 7.5 bits per character (Figures 4 and 5).

The entropy analysis of the first sample of Tango produced a very high change varying in the range of 3.5 and 7.0 bits per character, and there were several high spikes with entropy of 7.9 bits per character (see Figure 4). However, for the second sample (see Figure 5), the entropy distribution was even in the range of 4.0 and 5.0 bits per character with few sudden spikes or drifts.

6. Discussion

Table 3 summarizes our experiment findings.

Skype text communications were found to be encrypted and Skype voice communications had no cluster in the histogram analysis with high entropy.

Google Talk text communications are encrypted. The voice communications did not have any cluster in the histogram analysis and the entropy results had gradual changes with no sudden rise or fall in entropy. This suggests that Google talk encrypts voice communications.

For ICQ, the text communications are encrypted. Clusters were found in the histogram analysis and the entropy was uneven for sample 1 and steady changes were observed in sample 2. These findings suggest that ICQ does not encrypt the voice communications.

Viber text communications were determined to be encrypted. The voice communications had cluster in histogram analysis and high fluctuation in entropy analysis for both set of data, which suggests that encryption is not used for voice communications.

For Nimbuzz, the text communications were determined to be encrypted. Clusters were found in histogram analysis but the entropy analysis showed steady changes. The clusters in the histogram analysis were found to be in similar regions. The steadiness of entropy results and clusters in the same region strongly indicates that Nimbuzz voice communications are encrypted.

We found Yahoo text communications sent by the user to be in plaintext. The histogram analysis of Yahoo VoIP did not show any cluster, although the entropy results had high fluctuations. The high fluctuations in entropy suggest that voice communications are not encrypted.

Fring text communications are encrypted. There were clusters found in the histogram analysis with high fluctuation of entropy in the entropy analysis. The results suggest that voice communications are not encrypted.

Vonage text communications were determined to be encrypted. Clusters in histogram analysis were found in different regions, and the entropy results were very consistent with few spikes. The entropy results suggest that voice communications are encrypted, but the histogram analysis suggest otherwise. Therefore, we believe that voice communications are encoded and not encrypted.

WeChat text communications are encrypted. There were clusters in the histograms, and uneven entropy distributions were observed in the entropy analysis. The results suggest that voice communications are not encrypted.

Finally, Tango text communications are encrypted. No clusters in histogram were found, and the entropy distribution for sample 1 had fluctuations, but sample 2 produced evenly distributed entropy. Findings from three of the four samples suggest that voice communications are encrypted.

7. Conclusion

We examined ten popular mVoIP applications (see Table 2), and determined that only Yahoo application does not encrypt text communications. Using both histogram and entropy analysis, we determined that Skype, Google Talk, Nimbuzz and Tango encrypt voice data; and ICQ, Viber, Yahoo, Fring, Vonage and WeChat use some sort of voice encoding mechanism, but does not encrypt the voice data. Our results contribute towards a better understanding of legal interception of mVoIP communications. Future work includes decoding the captured unencrypted sessions to determine what user information (e.g. number of the mobile handset, location, IMEI) is being transmitted by the mVoIP applications during the login and voice or text communication session.

References

[1] Australian Government Dapertment of Broadband Communications and Digital Economy, "Statistical Snapshot", 2013.

[2] Menghui, Y., Hua, L., and Tonghong, L., "Implementation and Performance for Lawful Intercept of Voip Calls Based on Sip Session Border Controller", Proceedings of the IEEE 10th International Conference on Computer and Information Technology (CIT), 2010, pp. 2635-2642.

[3] Vrakas, N., and Lambrinoudakis, C., "An Intrusion Detection and Prevention System for Ims and Voip Services", International Journal of Information Security, 2(3), 2013, pp. 201-217.

[4] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E., "Sip: Session Initiation Protocol", Rfc 3261, Internet Engineering Task Force (IETF) Network Working Group.

[5] http://www.itu.int/rec/T-REC-H.323/, accessed 19 April 2013.

[6] http://tools.ietf.org/wg/p2psip/, accessed 19 April 2013, 2013.

[7] Keromytis, A.D., "A Comprehensive Survey of Voice over Ip Security Research", IEEE Communications Surveys & Tutorials, 14(2), 2012, pp. 514-537.

[8] Wang, X., Chen, S., and Jajodia, S., "Tracking Anonymous Peer-to-Peer Voip Calls on the Internet", Proceedings of the 12th ACM Conference on Computer and Communications Security, 2005, pp. 81-91.

[9] Chen, S., Wang, X., and Jajodia, S., "On the Anonymity and Traceability of Peer-to-Peer Voip Calls", IEEE Network, 20(5), 2006, pp. 32-37.

[10] Vrakas, N., Geneiatakis, D., and Lambrinoudakis, C., "A Call Conference Room Interception Attack and Its Detection": Trust, Privacy and Security in Digital Business, Lecture Notes in Computer Science, 2010, pp. 38-44.

[11] Wang, X., Zhang, R., Yang, X., Jiang, X., and Wijesekera, D., "Voice Pharming Attack and the Trust of Voip", Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks (Securecomm), 2008, pp. 1-11.

[12] Zhang, R., Wang, X., Farley, R., Yang, X., and Jiang, X., "On the Feasibility of Launching the Man-in-the-Middle Attacks on Voip from Remote Attackers", Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, 2009, pp. 61-69.

[13] Zhang, R., Wang, X., Yang, X., and Jiang, X., "On the Billing Vulnerabilities of Sip-Based Voip Systems", Computer Networks, 54(11), 2010, pp. 1837-1847.

[14] Koo, T., and Lin, C., "Voip Interception in P2P Sip Environment", Proceedings of the 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010, pp. 331-334.

[15] Seedorf, J., "Lawful Interception in P2P-Based Voip Systems", Principles, Systems and Applications of Ip Telecommunications Services and Security for Next

Generation Networks, Lecture Notes in Computer Science, 2008, pp. 271-235.

[16] Milanovic, A., Srbljic, S., Raznjevic, I., Sladden, D., Matosevic, I., and Skrobo, D., "Methods for Lawful Interception in IP Telephony Networks Based on H.323", Proceedings of the IEEE Region 8 Computer as a Tool (EUROCON), 2003, pp. 198-202.

[17] Verscheure, O., Vlachos, M., Anagnostopoulos, A., Frossard, P., Bouillet, E., and Yu, P.S., "Finding "Who Is Talking to Whom" in Voip Networks Via Progressive Stream Clustering", Proceedings of the IEEE Sixth International Conference on Data Mining, 2006, pp. 667-677.

[18] Takahashi, T., and Lee, W., "An Assessment of Voip Covert Channel Threats", Proceedings of the Third International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm), 2007, pp. 371-380.

[19] Srivatsa, M., Iyengar, A., and Liu, L., "Privacy in Voip Networks: A K-Anonymity Approach", Proceedings of the IEEE 28th Conference on Computer Communications (INFOCOM), 2009, pp. 2856-2860.

[20] Freire, E.P., Ziviani, A., and Salles, R.M., "Detecting Voip Calls Hidden in Web Traffic", IEEE Transactions on Network and Service Management, 5(4), 2008, pp. 204-214.

[21] Gomes, J., Inacio, P., Pereira, M., Freire, M., and Monteiro, P., "Identification of Peer-to-Peer Voip Sessions Using Entropy and Codec Properties", IEEE Transactions on Parallel and Distributed Systems, 2012, pp. 1-11.

[22] Wang, C.-H., and Liu, Y.-S., "A Dependable Privacy Protection for End-to-End Voip Via Elliptic-Curve Diffie-Hellman and Dynamic Key Changes", Journal of Network and Computer Applications, 34(5), 2011, pp. 1545-1556.

[23] Azab, A., Watters, P., and Layton, R., "Characterising Network Traffic for Skype Forensics", Proceedings of the Third Cybercrime and Trustworthy Computing Workshop (CTC), 2012, pp. 19-27.

[24] https://support.skype.com/en/faq/FA31/does-skype-use-encryption, accessed 22 April 2013.

[25] Blond, S.L., Zhang, C., Legout, A., Ross, K., and Dabbous, W., "I Know Where You Are and What You Are Sharing: Exploiting P2p Communications to Invade Users' Privacy", Proceedings of the ACM Internet Measurement Conference (SIGCOMM), 2011, pp. 45-60.

[26] Sengar, H., Zhen, R., Haining, W., Wijesekera, D., and Jajodia, S., "Tracking Skype Voip Calls over the Internet", Proceedings of the IEEE 29th Conference on Computer Communications (INFOCOM), 2010, pp. 1-5.

[27] King, A., and Lyons, K., "Automatic Status Updates in Distributed Software Development", Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering, 2011, pp. 19-24.

[28] Saint-Andre, P., "Extensible Messaging and Presence Protocol (Xmpp): Core", Rfc 6120, Internet Engineering Task Force (IETF), 2011

[29] http://xmpp.org/extensions/xep-0166.html, accessed 15 May 2013.

[30] http://www.bigblueball.com/im/googletalk/, accessed 30 April 2013.

[31] Jahanirad, M., Al-Nabhani, Y., and Noor, R.M., "Security Measures for Voip Application: A State of the Art Review", Scientific Research and Essays, 2011, pp. 4950-4959.

[32] http://www.viber.com/, accessed 17 April 2013.

[33] Appelman, M., Bosma, J., and Veerman, G., "Viber Communication Security: Unscramble the Scrambled", 2011.

[34] http://www.bkav.com/top-news/-/view_content/content/46264/critical-flaw-in-viber-allows-full-access-to-android-smartphones-bypassing-lock-screen, accessed 30 April 2013.

[35] http://www.icq.com/legal/privacypolicy/en, accessed 16 April 2013.

[36] http://au.messenger.yahoo.com/features/, accessed 14 August 2013.

[37] http://www.voiceofgreyhat.com/2012/07/yahoo-voice-compromised-450k-login.html, accessed 19 April 2013.

[38] http://www.nimbuzz.com/en/support, accessed 28 April 2013.

[39] http://www.voipusersconference.org/2011/jabber-jitsi-nimbuzz/, accessed 28 April 2013.

[40] http://www.fring.com/, accessed 10 May 2013.

[41] http://www.tango.me/, accessed 10 May 2013.

[42] http://www.vonagemobile.com/, accessed 12 may 2013.

[43] http://www.wechat.com/en/, accessed 15 May 2013.

[44] http://www.willhackforsushi.com/code/pcaphistogram.pl, accessed 10 May 2013.

[45] Shannon, C.E., "Prediction and Entropy of Printed English", Bell Systems Technical Journal, 30(1), 1951, pp. 50-64.

[46] https://github.com/batidiane/pyNetEntropy, accessed 10 May 2013.