IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 2 (Jan. - Feb. 2013), PP 01-09
www.iosrjournals.org

Attack Determination and its Security Analysis for Personal Communication in VoIP Networks

Dr.K.Venkatachalapathy[1], R.Janani[2]
[1]*Department of Computer Science and Engineering, Annamalai University, Chidambaram- 608 001, India*
[2]*Department of computer Science and Engineering, Annamalai University, Chidambaram- 608 001, India*

Abstract: *Voice over Internet Protocol (VoIP) is a technology that enables one to make and receive calls through the Internet instead of using the traditional analog PSTN (Public Switched Telephone Network) lines. Gtalk and Skype are frequently used for this purpose, which is a third party. So, we are utilizing their environment for conferencing. We are in need of a server to transfer the information from source to destination i.e. we are compromising our privacy at a certain level as they do not provide full security to our audio packets. Here, we do propose, novel flow analysis attacks that ensures both privacy and demonstrates the vulnerabilities in peer-to-peer VoIP networks. Solutions are proposed by quantifiable k-anonimity metrices.*
Keywords: *Flow analysis attacks, k-anonymity, mix networks, privacy, VoIP networks*

I. Introduction

With the most major telecommunications carriers currently in process of realying Voice-over-IP (VoIP) services for mass deployment, it's clear that IP telephony is finally headed for prime time. However, the promise of mass VoIP consumption also increases the risk of widespread security violations, spawning a new sense of urgency to fill in potential security gaps now before hackers wreak havoc on corporate voice networks. Until now, VoIP security has been easily overshadowed by the attractiveness of this new technology and the extensive features it promises to provide. Security hasn't been a particularly critical subject. Since in the past, most IP voice traffic remained on local and wide area enterprise networks, which were more or less secure and protected from the public Internet. But as VoIP usage is becoming widespread and Internet telephony is coming into play, enterprises and home users are becoming subject to the same security risks that have affected data networks for decades, thus opening the door to a whole new realm of security risks. This is largely due to the fact that next-generation Voice networks are IP-based and all IP protocols for sending voice traffic contain flaws.

An Internet environment can be considered particularly hostile for VoIP deployments for a number of reasons. Most important is that attacks are not traceable and the whole network is exposed to all public spoofing and sniffing. There was never been enough identification of the potential vulnerability to danger of devices communicating on the Internet makes security threats commonplace. This signifies that any VoIP device communicating insecurely in an Internet environment is at the risk of security breaches. Voice over Internet Protocol (VoIP) is a technology that enables one to make and receive calls through the Internet instead of using the traditional analog PSTN (Public Switched Telephone Network) lines. VOIP technology converts the analog telephone communication signals into digital communication signals and transfers through the data networks it may be a wide area network, local area network or the internet otherwise we can say that the sound is recorded and converted to computer data and transferred through internet to the destination where it again converted back from digital to analog sound which can be heard using speakers or headphone. In VoIP, the sound is converted into data packets and transferred to destination through the third party server. They may ensure the security up to certain level but taking into account the part of privacy, we could not guarantee it. The confidential discussions could not be carried out over the VoIP networks. In order to overcome these disadvantages we are implementing a peer to peer VOIP personal network setup.

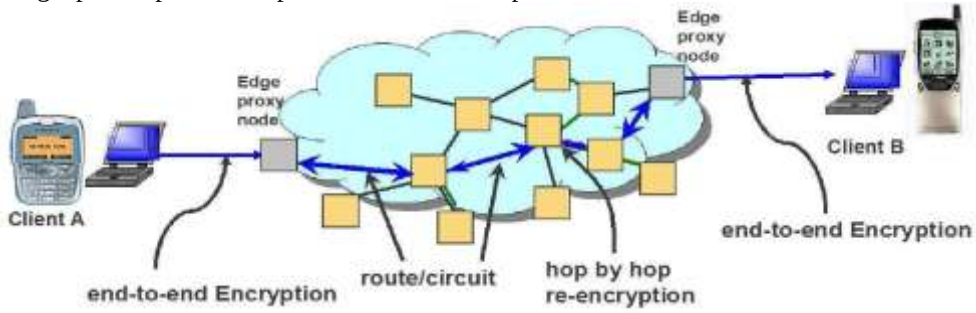

Fig.1. Anonymizing VoIP network

This paper examines anonymity for QoS sensitive applications on mix networks using peer-to-peer VoIP service as a sample application. In Fig. 1 shows a peer-to-peer VoIP network typically consists of a core proxy network and a set of clients that connect to the edge of this proxy network. This network allows a client to dynamically connect to any proxy over the network and to place voice calls to other clients over the network.

The following portions of this paper are organized as follows: we present a reference model for a VoIP network followed by flow analysis attacks in Section 3. We sketch an implementation of our proposal and present experimental results that quantify the performance in section 4.

II. VoIP ROUTE SETUP PROTOCOL

VoIP traffic can be classified into call signaling, call control, and media communications. Depending on the VoIP protocol and policies used, these communications may use either one channel or many different channels. Channels are TCP/UDP connections between two network elements. From a security point of view, all of these connections may need to be secured, i.e. authenticated and encrypted. Some of the mechanisms that may provide security in a VoIP environment are:

- Authorization
- Authentication
- Transport Layer Security (TLS)
- Media encryption (SRTP)

VoIP call signaling and call control can be secured by implementing some form of Authorization, Authentication or Transport Layer Security (TLS/SSL) mechanism.

2.1 Authorization

Authorization implies that the devices might be configured in such a way to allow traffic from only a select group of IP addresses. This mechanism shields the device to an extent from denial-of-service attacks.

2.2 Authentication

Authentication may require two communicating VoIP devices each other before the actual communication begins. This mutual authentication might be based on a shared secret that is known prior to the communication, making it difficult if not impossible for an attacker to masquerade identities.

2.3 Transport Layer Security

Transport Layer Security (TLS) can provide a secure communication channel between two communicating entities. The primary goal of the TLS Protocol is to provide privacy and data integrity between two communicating applications. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. A device incorporating TLS can be configured to allow only secure SIP signaling with other devices. This mandates that the client first sets up a TLS/SSL connection to the server and then exchanges encrypted SIP messages with it on the secure connection. Since this secure communication is based on a shared secret known only to the server and the client, this mechanism makes it very difficult, and again perhaps impossible, for an eavesdropper to view, manipulate, or replay the messages exchanged.

2.4 SRTP

Media communications can also be secured by incorporating some form of encryption mechanisms. VoIP phones may encrypt audio streams via SRTP (Secure Real-time Transport Protocol). SRTP is a security profile for RTP that adds confidentiality, message authentication, and replay protection to that protocol. SRTP is ideal for protecting Voice over IP traffic because it can be used in conjunction with header compression and has no effect on IP Quality of Service. It creates a unique key stream for each RTP packet, therefore making it almost impossible for eavesdroppers to retrieve the original RTP stream from the encrypted SRTP stream. SRTP also provides replay protection, which is undoubtedly important for multimedia data. Without replay protection it would be possible for an adversary to perform simple manipulations on data and subvert security.

For example, in a voice application, the phrase “yes” could be substituted for “no” if replay protection is not present. SRTP achieves high throughput and low packet expansion by using fast-stream ciphers for encryption, an implicit index for synchronization, and universal hash functions for message authentication. SRTP proves itself to be a suitable choice for the most general scenarios as well as the most demanding ones. The main security goals of SRTP are to ensure the confidentiality of the RTP payload, the integrity protection of the entire RTP packet (including protection against replayed RTP packets), and implicit authentication of the header by using ‘seek able’ stream ciphers, SRTP avoids the denial of service attacks that are possible on stream ciphers that lack this property.

Fig.2. VoIP – COMPLEX SERVICE

The connection made is setup avoiding the third party server, as they can interrupt our voice communication which spoils privacy. The Peer-Peer network setup is necessary in order to avoid the third party server involvement. The work of this connection is to read up the whole network structure when is being connected since our system too acts like a proxy server over the network. This produce the list of peers that were available aver the network which is suitable for establishing the connection.

The connection gets all peers that are available for establishing connection for making a privacy calling. The call must be forwarded only for the particular peer that the user in the particular peer decides or wish to make/accept a call. The protocol operates in four steps: *initsearch* (initiates a route setup by *src*), *processSearch (*process route setup request at some node), *processResult* (process results of a route setup request at some node), *finSearch* (concludes the route setup). One should note that flow analysis attacks exploits only the shortest path property and are *independent* of the concrete route setup protocol.

2.4.1 initsearch

A VoIP client *src* initiates a route setup for a receiver *dst* by broadcasting search (searchId, sipurl = *dst*.sipurl, ts = curTime) to all nodes p E ngh(*src*), where ngh(*src*) denotes the neighbors of node src in the VoIP network. Each VoIP client is identified by a URL (say, sip@example.com). The search identifier searchId is a long randomly chosen unique identifier denotes the time stamp at which the search requst was initiated.

2.4.2 processSearch

Let us suppose p receives search (searchId, sipurl, ts) from its neighber q. If p has seen searchId in the recent past, then it drops the search request. Otherwise, p checks if sipurl is the URL of a VoIP client connected to p. If yes, p returns its IP address using result(searchId,p) to q. P broadcast search (searchId, sipurl,ts) to all P' E ngh(p)-{q} and caches the search identifier <searchId,sipurl,q> in its recently seen list. Note that p' has no knowledge of where the search request is initiated.

2.4.3 processResult

Let us suppose p receives result (searchId, q) from q. Note that p has no knowledge as to where the search result was initiated. P looks up its cache of recently seen search queries to locate <searchId,sipurl,prev>. p adds a routing entry <sipurl,q> and forwards result(searchId,p) to prev.

2.4.4 finSearch

When *src* receives result (searchId,q) from q, it adds a routing entry <*dst*,q> to its routing table. The route setup protocol establishes the shortest overlay network route between *src* and *dst.* This observation follows from the following facts:

1) The first search request that reaches a node p must have traveled along the shortest route from *src* to p, and
2) In *processSearch*, a node p records the neighbor q through which it received the first search request. This indicates that the shortest route from src to p is via q.

   Setting *p = dst* shows that the route setup procedure in *processResult* builds the shortest VoIP network path from *src* to *dst.*

After a successful route setup, the clients' *src* and *dst* exchange an end-to-end media encryption key and switch to the media delivery phase. The media delivery phase additionally uses hop-by-hop re-encryption using pair-wise shared keys between neighboring proxy nodes in the VoIP network.

An external observer tapping into the VoIP network may observe <*src*IP; *dst*IP; *src*Port; *dst*Port, EKp,q (EKsrc,dst (media))>, where Kp,q denotes a pair-wise shared symmetric key between neighboring nodes p and q on the route, K*src,dst* denotes the end- to-end encryption key, and *media* denotes an encoding of media bits.

The encryption could be carried out using various methods. Probabaly usage of DES, but its security is compromised and is out of date. Triple DES contains multiple loops within it and could cause time delay which affects the end-users performance. We use AES algorithm, which is moderate than DES and comprices high performance compared to Triple DES for encrypting a voice packets at the level of end-to-end media encryption.

2.5 AES Algorithm

AES has a fixed block size of 128 bits and a key size of 128, 192 or 256 bits. AES operates on a 4×4 array of bytes termed the state. For encryption, each round of AES (except the last round) consists of four stages:

2.5.1 Add Round Key -- each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule.

2.5.2 Sub Bytes -- a non-linear substitution step where each byte is replaced with another according to a lookup table.

2.5.3 Shift Rows -- a transposition step where each row of the state is shifted cyclically a certain number of steps.

2.5.4 Mix Columns -- a mixing operation which operates on the columns of the state, combining the four bytes in each column using a linear transformation.

The final round replaces the Mix Columns stage with another instance of Add Round Key.

III. Flow Analysis Attacks

In this section, we describe flow analysis attacks on VoIP networks. These attacks exploit the shortest path nature of the voice flows to identify pairs of callers and receivers on the VoIP network. Similar to other security models for VoIP networks, we assume that the physical network infrastructure is owned by an untrusted third party (say, tier one/two network service provider). Hence, the VoIP service must route voice flows on the untrusted network in a way that preserves the identities of callers and receivers from the untrusted network.

We assume that the untrusted network service provider (adversary) is aware of the VoIP network topology, and the flow rates on all links in the VoIP network. The network service provider can obtain VoIP topology and flow information using traffic analysis (see Fig. 3) or using various measurement-based approaches (such as expanding ring search on the network topology). We experimentally show that the attack can be very effective even when only one-third of the links are monitored by the adversary.

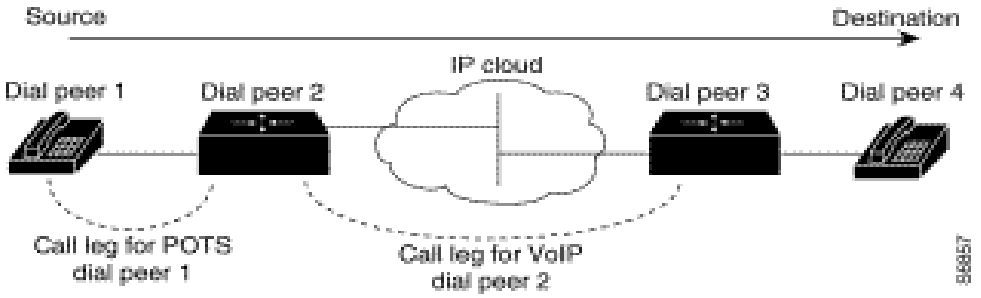

Fig.3. Call flow in network

An important factor to consider before starting any VoIP troubleshooting or debugging is that VoIP calls are made up of three call legs. These call legs are source Plain Old Telephone Systems (POTS), VoIP, and destination POTS. This is shown in this diagram. Trouble shooting and debugging needs to be first focused on each leg independently and then on the VoIP call as a whole.

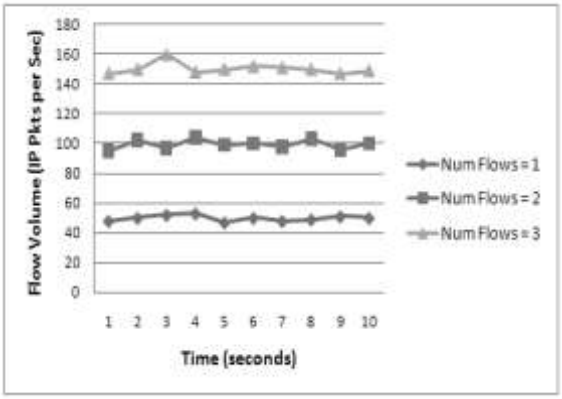

Fig.4. Inferring number of flows from flow volume

3.1 Shortest path algorithm

We are using the shortest path algorithm in order to send the data packets from the caller to the receiver because it reduces the time and cost of making the packets to travel from source to destination.The Shortest path algorithm is used for finding the shortest path that is available between the two nodes i) The Source and ii) The Destination. There are multiple nodes that are available over the network that was scanned through the route setup. Now the connection has to be established between the source and the destination. The Shortest path Algorithm (the Dijkstra's algorithm) is being used for this purpose.

3.1.1 The Dijsktra's Algorithm Pseudocode:

The code u := vertex in *Q* with smallest dist[], searches for the vertex *u* in the vertex set *Q* that has the least dist[*u*] value. That vertex is removed from the set *Q* and returned to the user. Distance between (*u*, *v*) calculates the length between the two neighbor-nodes *u* and *v*. The variable *alt* on lines 15 & 16 is the length of the path from the root node to the neighbor node *v* if it were to go through *u*. If this path is shorter than the current shortest path recorded for *v*, that current path is replaced with this *alt* path. The previous array is populated with a pointer to the "next-hop" node on the source graph to get the shortest route to the source.

1. **function** Dijkstra(*Graph*, *source*):
2. **for each** vertex *v* in *Graph*:
3. dist[*v*] := infinity ;
4. previous[*v*] := undefined ;
5. **end for**
6. dist[*source*] := 0 ;
7. *Q* := the set of all nodes in *Graph* ;
8. **while** *Q* **is not** empty:
9. *u* := vertex in *Q* with smallest distance in dist[] ;
10. remove *u* from *Q* ;
11. **if** dist[*u*] = infinity:
12. break ;
13. **end if**
14. **for each** neighbor *v* of *u*:
15. *alt* := dist[*u*] + dist_between(*u*, *v*) ;
16. **if** *alt* < dist[*v*]:
17. dist[*v*] := *alt* ;
18. previous[*v*] := *u* ;
19. decrease-key *v* in *Q*;
20. **end if**
21. **end for**
22. **end while**
23. **return** dist;

Fig. 5 illustrates the Dijkstra's algorithm search for finding path from a start node (*src* node) to End node (*dst* node) in a robot motion planning problem.

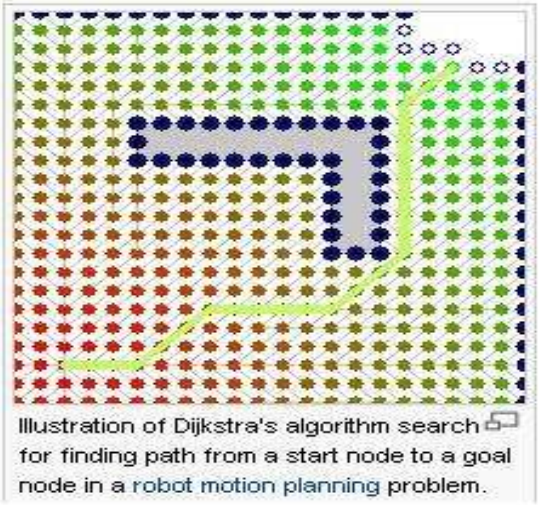

Fig.5. Illustration of Dijkstra's algorithm search

3.2 NAÏVE Tracing Algorithm

This algorithm is used for finding out the alternate path other than the shortest path for voice dispersion through nodes in order to send the packets without affected and to avoid collision while the packets are transferred mainly it is used to take care of traffic problems. Let us consider sample topology that contains the label on edges of topology indicates the number of voice flows. A trace from caller p1 results in p1=p2=p3=p4=p5=1. Filtering out the VoIP proxy nodes (p5) and the caller (p1), the clients' p2, p3 and p4 could be potential destinations for a call emerging from p1. The naive tracing algorithm doesn't consider the shortest path route it will help to get the alternate longest route path [5].

TRACE(Graph G=hV , Ei, Caller *src*)

1. **for each** vertex v 2 V
2. f[v] = 0; label[v] = false
3. **end for**
4. f[*src*] = 1; label[*src*] = true
5. **while** pick a vertex v labeled true
6. label[v] = false
7. **for each** node u such that (u, v) 2 E
8. **if** (f[u] = 0)
9. f[u] = 1; label[u] = true
10. **end if**
11. **end for**
12. **end while**

3.3 Flow Analysis Algorithm

In this section, we use the flow measurements to construct a probability distribution over the set of possible receivers. Let G□ be a subgraph of obtained using the top-*k* shortest path algorithm with caller *src*. Let $nf(p \rightarrow q)$ denote the number of flows on the edge $p \rightarrow q$. Let *in(p)* denote the total number of flows into the node p. Note that both $nf(p \rightarrow q)$ and *in(p)* are observable by an external adversary. Assuming a node p in the VoIP network performs perfect mixing, the probability that some incoming flow is forwarded on the edge $p \rightarrow q$ as observed by an external adversary is $\frac{nf(p \rightarrow q)}{in(p)}$.

Let *f(p)* denote the probability that a VoIP flow originating at *src* flows through the node *p*. The function *f* is recursively defined on the directed edges in G□= □V □, E□> as follows:

$$f(q) = \sum_{p \rightarrow q \in E^K} f(p) * \frac{nf(p \rightarrow q)}{in(p)}, \quad (1)$$

With the base cases *f* (*src)* = 1 and *in(src)* = 1. Now, every VoIP client p (p ≠ *src)* is a possible destination for the VoIP flow originating from *src* if *f(p)* > 0.

3.4 Compromised Proxies

In addition to passive observation-based attacks, the adversary could actively compromise some of the nodes in the VoIP proxy. We assume an honest-but-curious model for the compromised nodes. A compromised node *p* reveals its mixing information (namely, $nf(r \rightarrow p \rightarrow q)$) to an adversary, where $nf(r \rightarrow p \rightarrow q)$ denotes the number of voice flows that were routed from r to q by the malicious node *p*. With slight abuse of notation, we use $f(p \rightarrow q)$ to denote the probability that a VoIP call from *src* traverses the edge $p \rightarrow q$. Hence, the new flow analysis equations are as follows:

$$f(p \rightarrow q) = \begin{cases} f(p) * \dfrac{nf(p \rightarrow q)}{in(p)}, & honest\ p, \\ \sum_{r \rightarrow p} f(r \rightarrow p) * \dfrac{nf(r \rightarrow p \rightarrow q)}{nf(r \rightarrow p)}, & malicious\ p \end{cases}$$

$$f(q) = \sum_{p \rightarrow q} f(p \rightarrow q). \qquad (2)$$

Compromised proxy nodes significantly enhance attack efficacy; for instance, when 20 percent of the nodes are compromised, the top-1 probability improves from 0.23 to 0.50.

3.5 K- Anonymity algorithm

K-Anonymity: A voice flow from *src* to *dst* is said to be k-anonymous if the size of a candidate receiver set identified by an adversary using the naïve tracking algorithm is no smaller than *k.*

K-anonymity algorithm is used for providing the anonymous network so that hackers could not identify the flow of packets over the network. let us consider the caller as S and the receiver as R the voice packets need to move through the personal network which consists of proxy nodes . The hacker can hack the voice easily from the proxy node which is present next to the caller or at the proxy node which is present before the destination R so we are using the k-anonymity algorithm which will hide the proxy nodes which is next to the caller and which is present before the destination.

IV. Performance Evaluation

We briefly describe an implementation of our algorithms using Phex[4]: an open source implementation of shortest route setup protocol. VoIP protocols operate on top of the peer-to-peer infrastructure. We have implemented our algorithms as pluggable modules that can be weaved into the Phex client code using AspectJ[12]. Our implementation is completely transparent to the VoIP protocol that operates on top of the peer-to-peer infrastructure. Also, our implementation does not require any changes to topology construction and , maintenance algorithms (as nodes join, leave, fail, or recover) and the underlying TCP/IP or UDP-based communication libraries. The Phex broadcast search protocol has four operations : init-Search, process-Search, process-Result, and fin-Search. These four operations are implemented as event handlers in Phex. When a Phex client receives a message, it determines the type of the message (search request, search result, etc.) and triggers the appropriate event handler.

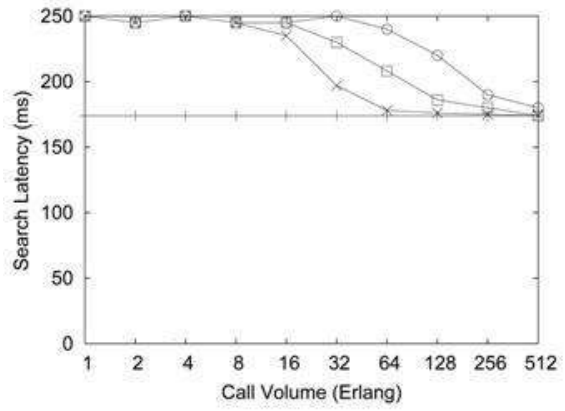

Fig.6. Search latency

Fig. 6 shows the latency of a search operation as we vary the call volume and the anonymity parameter k (using c = 1). This experiment shows that incur about 30- 40 percent higher search latency. One should note that the

search latency only affects the initial connection setup time. Once the route is established, it ensures good quality voice conversations by limiting the path latency to 250 ms.

Fig. 7 shows the average number of concurrent VoIP calls handled by a node in the VoIP network (using c = 1).

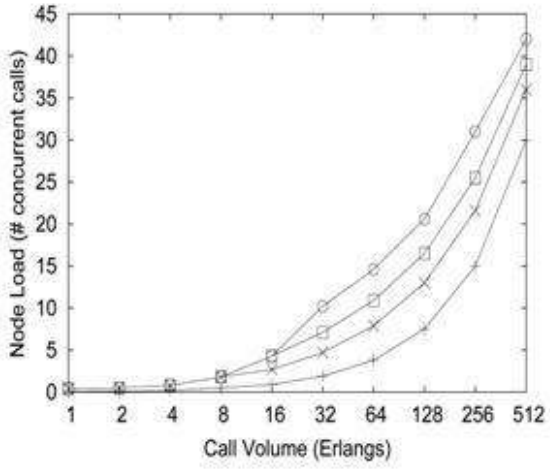

Fig.7. Node load

We can increases the average number of proxies that route one voice call, and thus increases the average load on a proxy. The percentile increase in average node load for higher call volumes is small.

V. Related Work

VoIP is not just another application running on the top of the IP infrastructure. VoIP is a complex service with its own business models and set of features offered to the end-user, similar to existing PSTN and PBX offerings. Over the years, service providers and PBX vendors have established their respective brands as being synonymous with high levels of reliability, quality and security and need to preserve these attributes in their VoIP offerings. The VoIP reliability requirements are very stringent and approaching 92.04% (5 minutes of downtime a year). Clearly, this level of reliability calls for automated, real time response to the security threats and attacks. The types of attacks that are common in the data security realm and may render email or the computer network unusable for several hours are not acceptable when it comes to IP communication.

VoIP characteristics such as high sensitivity to Quality of Service (QoS) parameters, real-time nature of the service, a wide range of infrastructure devices, protocols and applications, and interaction with the existing phone networks require different techniques and methodologies that will support PSTN like security and reliability. VoIP QoS sensitivity to packet delay, packet loss, and packet jitter makes most of the existing security solutions inadequate. Existing firewalls cannot efficiently handle new VoIP protocols such as standard based SIP and a wide range of vendor proprietary protocols since they rely on dynamic port ranges and do not support Network Address Translation (NAT) very well. A new generation of the firewalls called Session Border Controllers(SBC) is addressing most of these problems.

Most of the firewalls, Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), and similar security devices rely on deep packet inspection techniques that introduce delays and jitter to the VoIP packet streams thus impacting overall QoS. In the VoIP world, maximum packet delay is set to 150 ms (in some cases higher), but the multi-layer nature of security infrastructure could add significant delays and jitter that would make the VoIP services unusable.

There is also the issue of balance between encryption and QoS. Existing encryption engines will introduce additional jitter and delay that would be cumulative due to hop-by-hop encryption schemas foreseen to be used by VoIP calls. For the foreseeable future PSTN and VoIP networks will coexist and require media gateways that provide internetworking between a carrier's IP network and TDM based PSTN networks. This could enable cross-network security attacks impacting existing PSTN networks.

VoIP is a real time service, i.e., it is happening in real-time and no information is stored anywhere on the network. As a result, any loss of information cannot be recovered or retransmitted. This makes the VoIP services very susceptible to worms and DoS attacks that could very easily disrupt voice communication. Also, the complex nature of VoIP infrastructure consisting of a wide range of components and applications such as telephone handsets, conferencing units, mobile units, call processors/call managers, gateways, routers, firewalls, and specialized protocols requires system level approach where security is built into all the infrastructure layers and coordinated via a centralized control center.

VI. Conclusion

In this paper, we have addressed the problem of providing privacy guarantees in peer-to-peer VoIP networks. To overcome that First, we developed a personal network without the involvement of third party which consists of proxy nodes through which the voice will be transferred to the receiver so that the privacy will be guaranteed. Secondly, the path in which data transferred will be a shortest path or any other alternate path which will be selected with the help of shortest path and naive tracing algorithm through which we can achieve security. Finally, we focus on technical feasibility of privacy attacks and defenses on VoIP networks, we are enhancing the concept of conference call in p2p VOIP personal network and e will inform the receiver about the call arrival through message even if the receiver is busy with any other works in online and the security will be enhanced to avoid others hiring the communication between caller and receiver.

References

[1] Mudhakar Srivatsa, Arun Lyengar,Ling liu, Hongbo jiang "Privacy in VoIP Networks: Flow Analysis Attacks and Defense "IEEE20ll.
[2] "The Network Simulator NS-2," http://www.isi.edu/nsnam/ns/,2010.
[3] "The Network Simulator NS-2: Topology Generation," http://www.isi.edu/nsnam/ns/ns-topogen.html , 2010'
[4] Phex Client," http://www.phex.org, 2010
[5] "Skype—The Global Internet Telephone Company," http://www.skype.com, 2010.
[6] "Telegeography Research," http://www.telegeography.com, 2010.
[7] GT-ITM: Georgia, Tech Internetwork Topology Models," http://www.cc.gatech.edu/projects/gtitm/ , 2010.
[8] M.J. Freedman and R. Morris, "Tarzan: A Peer-to-Peer Anonymizing Network Layer," Proc. Ninth ACM Conf. Computer and Comm. Security (CCS), 2002.
[9] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second Generation Onion Router," Proc. 13th USENIX Security Symp., 2000.
[10] C# Network Programming book by Richard Blum.
[11] D.L. Donoho, A.G. Flesia, U. Shankar, V. Paxson, J. Coit, and S.Staniford, "Multiscale Stepping Stone Detection: Detecting Pairs of Jittered Interactive Streams by Exploiting Maximum Tolerable Delay," Proc. Fifth Symp. Recent Advances in Intrusion Detection(RAID), 2002.
[12] Eclipse. Aspectj Compiler," http://eclipse.org/aspectj , 2010.