On the Anonymity and Traceability of Peer-to-Peer VoIP Calls

Shiping Chen, Xinyuan Wang, and Sushil Jajodia, George Mason University

Abstract

Voice over Internet Protocol is a technology that enables people to use the Internet, rather than the traditional public switched telephone network, as the transmission medium for voice communications. VoIP is becoming increasingly popular due to its significant advantages in cost and flexible features compared with the plain old telephone system. The proliferation of VoIP calls has significant implications on the security and privacy aspects of voice calls. For example, the use of VoIP has made it much easier to achieve confidentiality and anonymity in voice communications. On the other hand, VoIP has imposed significant new challenges in providing the same call-identifying and wiretapping capabilities as those that exist in traditional circuit-switched networks. In this article we examine the privacy and security aspects of peer-to-peer (P2P) VoIP calls and show how the use of VoIP has substantially shifted the previous balance between privacy and security that exists in traditional PSTN calls. In particular, we show that the use of strong encryption and available low-latency anonymizing network at the same time does not necessarily provide the level of anonymity to VoIP that people would intuitively expect.

VoIP is a technology that enables people to use the Internet, rather than the traditional Public Switched Telephone Network (PSTN), as the transmission medium for voice communications. Because VoIP offers significant cost savings while providing more flexible and advanced features than the Plain Telephone System (POTS), more and more voice calls are now carried at least partially via VoIP. It has been estimated that VoIP will account for 75 percent of voice services worldwide by 2007 and the IP-based Public Branch Exchange (PBX) market will grow to $16 billion worldwide by 2006 [1]. The proliferation of VoIP calls, however, has introduced significant implications on the security and privacy aspects of phone calls.

When people talk over the phone, privacy is usually one of their top concerns. For example, people may want to keep the phone conversation confidential such that no eavesdropper could obtain the content of the conversation. In addition, people sometimes may want to keep their phone conversation anonymous such that no one else would know whom they have talked to or who has called them.

Law enforcement agencies (LEAs), on the other hand, need the capability to conduct lawful electronic surveillance to fight crime and terrorism. For example, LEAs are interested in knowing at what time calls were made, who has called the surveillance target, and to whom the surveillance target has called. The Communication Assistance for Law Enforcement Act (CALEA), enacted in 1994, requires telecommunication carriers and equipment vendors to provide wiretapping capability for LEAs to intercept communications and collect call-identifying information about calls to and from the surveillance targets.

In POTS, every telephone port attached to the PSTN has a unique phone number, and the voice path between the caller and callee of a call is set up and circuit-switched by the telephone switch. Unless some special crypto devices are used at both ends of the phone call, the conversation is transferred across PSTN without encryption. Therefore, all the call identifying information (i.e., caller number and callee number) and the call content are readily available to the telephone switch. It is fairly straightforward for the telephone switch to provide the call-identifying and wiretapping capabilities required by CALEA. On the other hand, it is difficult, at least for normal customers, to make the content of phone conversation confidential in POTS. It is even more challenging to achieve anonymity for PSTN phone calls.

The recent proliferation of VoIP has drastically changed the technical feasibility of achieving the security and privacy of phone calls. In contrast to PSTN calls, VoIP calls are packet-switched over the public Internet. The area code of the calling number or the called number does not necessarily correspond to the physical location of the caller or callee. In fact, the VoIP service provider Vonage (www.vonage.com) allows its customers to freely choose any area code no matter which state they are in. For computer to computer VoIP calls (e.g., www.skype.com), there is no phone number at all. In addition, the VoIP traffic could easily be encrypted by using the Secure Real-Time Transport Protocol (SRTP) [2] or publicly available encryption software, and be transformed by various IP level transformations such as IPsec/VPN tunnel [3] L2TP [4], NAT [5], and low-latency anonymizing networks (e.g., Onion Routing [6], Tor [7], Freedom [8], and Tarzan [9]). All these have made it much easier to achieve confidentiality and anonymity in VoIP calls. At the same time, it becomes extremely difficult to have the same call-identifying and wiretapping capabilities on VoIP calls as those that exist on traditional PSTN calls.

While VoIP has made it much easier to achieve confidentiality and anonymity in voice calls, LEAs are concerned that

0890-8044/06/$20.00 © 2006 IEEE

VoIP may become a "haven for criminals, terrorists, and spies" [10]. In a letter to the FCC [11], several federal law enforcement agencies considered the capability of tracking VoIP calls "of paramount importance to the law enforcement and the national security interest of the United States."

How to balance people's need for privacy and anonymity and LEAs' need for lawful electronic surveillance has been controversial. In this article, we leave the controversy aside and focus on the technical side of the anonymity and traceability aspects of VoIP calls. We examine techniques that could be used for achieving anonymity and confidentiality for VoIP calls, and their effectiveness and limitations. We then review techniques that could be used for tracking VoIP calls on the Internet and show that, under certain conditions, VoIP calls can be effectively tracked even if they have been encrypted end-to-end and anonymized through low-latency networks. Finally, we present our conclusions.

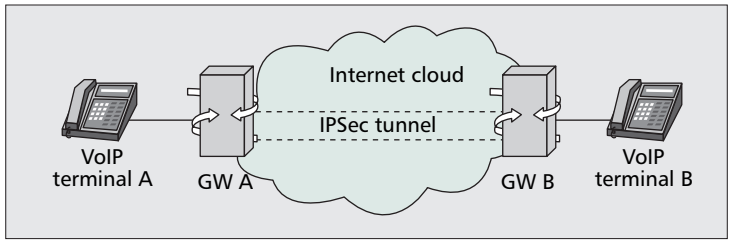
Anonymizing VoIP Calls

Conceptually, anonymity refers to the absence of a true or real identity. In the context of VoIP, there exist variations of anonymity. For example, at one time, a caller (or callee) may want to remain anonymous so that the other side does not know whom he/she is talking to; at other times, the caller and callee know to whom they are talking, but want their phone conversation to be anonymous to the third party so that no other people know that they are talking (or have talked) over the phone. In this article we consider the latter case about how to make VoIP calls anonymous to the eavesdropper who has access to the VoIP flows.

To achieve the anonymity of VoIP calls in the presence of an eavesdropper, it is necessary to conceal the content of the VoIP conversation. This can be achieved by using end-to-end encryption on the VoIP flow or using SRTP [2]. However, making the VoIP content confidential itself does not necessarily make the VoIP calls anonymous. For example, for an encrypted VoIP call between A and B, if the eavesdropper could somehow determine the IP addresses of A and B and associate them to one VoIP flow, he would know that A and B are talking (or have talked) over VoIP. Therefore, in order to make VoIP calls anonymous, the correspondence between the IP addresses of the caller and the callee must be concealed.

Existing VoIP services can be broadly classified into two categories: *managed* VoIP and *unmanaged* VoIP. Residential VoIP services and corporate VoIP services are typical managed VoIP, and direct computer-to-computer VoIP is a typical unmanaged VoIP. In managed VoIP, all the calls are set up and managed by some service provider through its service gateways. Unmanaged VoIP calls, however, are not set up and managed by a service provider, and they can use peer-to-peer technology to setup and route the calls (e.g., Skype calls). In managed VoIP, customers are usually assigned unique phone numbers as that exist in the traditional PSTN system, and they use VoIP phones (or a traditional phone behind VoIP adaptor) to make and receive calls. Users of unmanaged VoIP normally use soft phones as terminals, and use some user IDs instead of phone numbers to identify the caller and callee in their VoIP calls.

Since the managed VoIP calls are set up and managed by some service provider, the service provider has all the information about the caller and callee of any VoIP calls between their customers. For example, the service provider would know the phone numbers of the caller and/or callee of a VoIP call made/received by its customers, even if the VoIP traffic has been encrypted from end-to-end and anonymized through anonymizing the network. This is in contrast to unmanaged VoIP calls, where there is no third party who knows all the call setup information. Therefore, it is easier to achieve anonymity for unmanaged VoIP calls. We focus on unmanaged VoIP calls in the remainder of this article.



■ Figure 1. *Anonymous VoIP call setup using an IPSec tunnel.*

Anonymizing VoIP Calls via IPsec

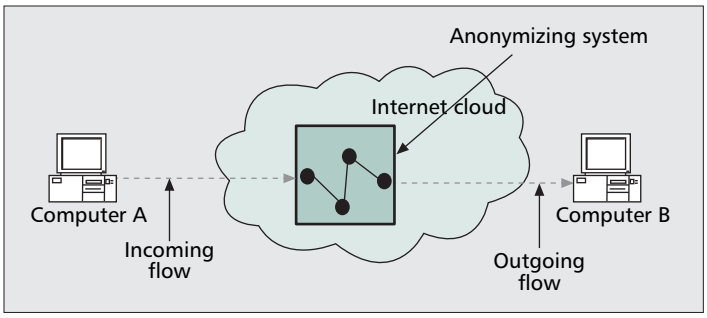
IPsec [3] is the standard security architecture that defines a suite of security protocols and encryption algorithms that provide various security services at the IP layer. IPsec defines two basic protocols: Encapsulating Security Payload (ESP) and Authentication Header (AH). AH provides connectionless integrity, data origin authentication, and an optional anti-replay service, whereas ESP provides confidentiality (encryption), and limited traffic-flow confidentiality in addition to connectionless integrity, data origin authentication, and anti-replay service. IPsec has been deployed widely to implement Virtual Private Networks (VPNs).

IPsec may operate in two different modes, depending upon whether the secure communication is between two endpoints directly connected or between two intermediate gateways to which the two endpoints are connected via a clear channel. The former is called transport mode, and the latter is called tunnel mode. ESP transport mode encrypts only the data portion (payload) of each packet, and leaves the original IP header untouched. Compared with transport mode, tunnel mode is more secure in the sense that it could conceal the original IP header in addition to the payload.

By using IPsec ESP tunnel mode, one could achieve partial anonymity for the VoIP calls. For example, we could construct an IPsec ESP tunnel and route VoIP calls through the tunnel. Figure 1 shows such a setting, where gateway A and B are connected by an IPsec VPN tunnel (using IPsec tunnel mode with ESP), user A routes its VoIP to B through gateway A, and user B routes its VoIP traffic to A through Gateway B. When VoIP flows pass through gateway A and B, the source and destination IP addresses would not be that of terminal A or B, but would be that of gateway A and B. Therefore, any eavesdropper between gateways A and B would not be able to identify the real IP addresses of the tunnelled VoIP flow. However, the VoIP flows between terminal A and gateway A and between terminal B and gateway B still show the IP addresses of terminals A and B as the source and destination IP addresses. Thus any eavesdropper between terminal A and gateway A or between terminal B and gateway B would be able to see the real IP addresses of VoIP flows between terminals A and B. Therefore, the IPsec tunnel could only make VoIP calls partially anonymous. Furthermore, if gateways A and B are known to belong to some organizations A and B, an eavesdropper between gateways A and B can infer that someone from organization A is talking to someone in organization B, which would reveal useful information about the conversation.

Centralized Anonymizing Systems

Anonymizing systems have been widely used to provide anonymity in data communication applications such as email and Web browsing. Based on their underlaying architectures,



■ Figure 2. *Traffic routed through a centralized anonymous system.*

anonymizing systems can be generally classified into two broad categories: centralized anonymizing systems and peer-to-peer anonymizing systems.

A centralized anonymizing system can be viewed as a black box with one or more entry and exit points. As shown in Fig. 2, one flow, when routed through the anonymizing network, would be broken into two different flows: an *incoming* flow and an *outgoing* flow. Instead of carrying their original source and destination IP addresses, the packets in the incoming flow will have the anonymizing system's entry point's IP address as their destination IP address, and the packets in the outgoing flow will have the anonymizing system's exit point's IP address as their source IP address. In this way, the incoming and the outgoing flows appear to be independent, since they carry different source-destination IP address pairs. When many incoming flows and many outgoing flows coexist at the same time, it is difficult for an outsider to link one incoming flow to its corresponding outgoing flow

A centralized anonymizing system could potentially consist of a sequence of proxy servers.[1] However, there is a trade-off between the anonymity achieved and the latency the system introduced. For those applications that are not in real time (e.g., email), the corresponding anonymizing system could afford large and variable latencies to achieve better anonymity. Examples of such large-latency anonymizing systems include Babel [12] and Mixminion [13]. For those real-time applications such as browsing, interactive sessions, and VoIP, the anonymizing system is constrained to have *low latency*. For example, VoIP has a stringent real-time requirement in that the total end-to-end delay should be no more than 150 ms. Examples of existing low-latency centralized anonymizing systems include Onion Routing [6], Tor [7], Freedom [8], anonymizer.com [14], and findnot.com [15]. However, most existing low-latency anonymizing systems are designed for protecting TCP or HTTP only, and they do not support UDP. Since VoIP systems are normally based on UDP, those TCP/HTTP-only low-latency anonymizing systems (such as Freedom and anonymizer.com) cannot be used directly for anonymizing VoIP calls. We have found that findnot.com is able to support any IP transport protocol, and we have successfully used findnot.com to anonymize Skype peer-to-peer VoIP calls without noticeable voice-quality degradation.

Compared with IPsec tunnel, centralized anonymizing systems achieve much better anonymity in that they conceal the correspondence between the sender and receiver by breaking the original flow into an incoming flow and an outgoing flow to and from the anonymizing system, respectively. Nowhere would the anonymized flow directly show the correspondence between the original sender and receiver. In order to identify the correspondence between the original sender and receiver, the incoming and outgoing flows have to be correlated.

[1] *Here we refer to the general proxy server rather than the SIP proxy server.*

Peer-to-Peer Anonymizing Systems

A peer-to-peer (P2P) network is a network where each node in the network acts as a peer and simultaneously functions as both "client" and "server" to the other nodes on the network. An anonymous P2P is a particular type of P2P network where all nodes are dynamically involved in anonymizing the network flows. Examples of existing anonymous P2P networks include Freenet [16], Entropy [17], GNUnet [18], and winny [19]. Tor is an anonymous P2P if all its nodes run in server mode.

Although all the existing anonymous P2P networks are designed for anonymous file sharing or anonymous publishing, it is possible to make VoIP calls anonymous by routing them through a specially designed anonymous P2P network. Figure 3 shows such a design, where each VoIP terminal (e.g., a computer with softphone installed) is a node in the anonymous P2P network, and all the nodes are connected via some protocol to form an overlay anonymous P2P network. Note that the network is not statically constructed, and all nodes can dynamically join and leave the network.

To make an anonymous VoIP call, the caller node can randomly choose a sequence of nodes and ask them to help to forward its traffic to the intended receiver node (callee node). For example, node A in Fig. 3 may choose nodes C and D to be its intermediate nodes[2] for forwarding its call to node B. Node B may use the same or a different set of nodes to forward its call traffic to A. To make the system more secure, they may change the route in the middle of conversation.

The philosophy behind the P2P VoIP anonymizing system is that each node on the network acts as a universal sender and universal receiver. This makes it difficult for an outsider to tell whether a node sends out a packet for itself or simply forwards it on behalf of another node. For example, node A in Fig. 3 acts as a caller as well as a forwarder for the call between nodes E and F (path A–C is shared by two calls) at the same time. An outsider is unable to determine if A is calling someone or just being a forwarder, as all the packets sent from A to C carry the same source and destination IP addresses. To further conceal the correspondence between the original packet sender and receiver, some nodes could send and receive bogus packets as cover traffic [9]. If the anonymizing P2P network contains many active nodes which are well connected, it would be extremely difficult for any outsider to identify the real source and destination of any packets observed.

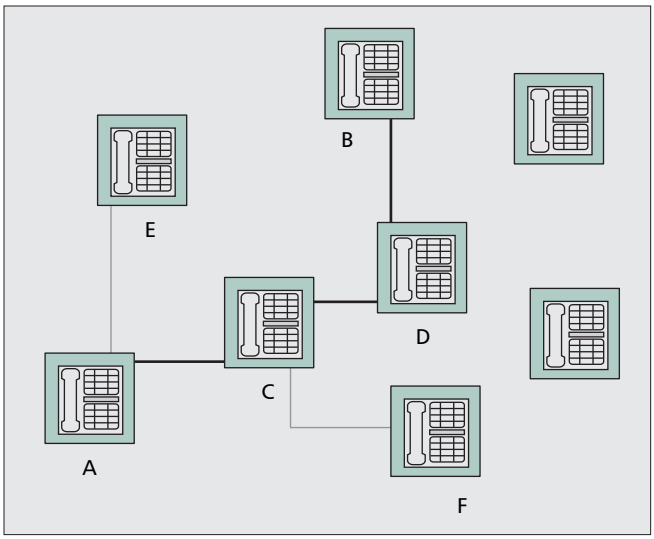
Summary of Anonymizing VoIP Calls

To make VoIP calls anonymous, it is necessary to conceal the content of the VoIP call, which can be achieved by SRTP or other end-to-end encryption techniques such as IPSec. However, concealing the content itself does not guarantee anonymity, and IPsec can only provide partial flow anonymity.

A centralized anonymizing system could provide reasonable anonymity by breaking the original flow into different flows with low latency. There are a number of centralized anonymizing systems commercially available and some of them are widely used by many people. In the next section, we will show that existing centralized low-latency anonymizing systems can be defeated by timing attacks.

P2P anonymizing systems (such as [9]) could achieve potentially better anonymity if there are enough nodes actively participating. When each node generates and receives cover traffic, it would be more difficult for any outsider to identify

[2] *Note that we assume that all the nodes in the example figure are connected at the IP level and are available to be used as intermediate nodes.*



■ Figure 3. *Anonymous VoIP call using P2P anonymizing network.*

the real sender and receiver of any packets. However, cover traffic would introduce additional overhead and waste useful bandwidth of the network.

Because the managed VoIP services are provided by some service provider, the service provider has all the information about calls made or received by it customers. It is very difficult for the customers of managed VoIP to achieve anonymity against a service provider, even if they use SRTP, strong end-to-end encryption, and all the existing anonymizing systems.

Tracing Anonymous VoIP Calls on the Internet

In this section we consider how the anonymous VoIP calls could be traced on the Internet. The goal of tracing anonymous VoIP is to effectively identify the caller and the callee of a particular VoIP call even if it is anonymized. Apparently the goal of tracing VoIP calls is in direct conflict with that of anonymizing VoIP calls. Here we leave the controversy between the anonymity and tracing aside, and instead focus on the technical feasibility on tracing anonymous peer-to-peer VoIP calls on the Internet.

In POTS, all the callers and callees are customers of some telephone service providers (e.g., AT&T, Verizon, etc.), and all calls are set up and switched by the telephone switches owned by one or more telephone service providers. Therefore, the telephone switch has all the call-identifying information of every call it has set up, and it is technically straightforward for the telephone switch to track any calls it has set up. In fact, existing call-identification of traditional PSTN calls is based on the signaling protocols that set up the calls. While signaling-protocol-based call-identification works fine for PSTN calls, it is difficult to be applied directly to VoIP calls due to the following reasons:

- The signaling protocols for VoIP calls are evolving.
- There are currently multiple competing VoIP signaling protocols (i.e., H.323 [20], MGCP [21], and SIP [22, 23]) and some of them is proprietary (i.e., the Skype signaling protocol)

In order to effectively identify VoIP calls, a new form of call-identifying information is needed. Ideally, the new form of VoIP call-identifying information should be applicable to all VoIP calls no matter what signaling protocols are used to set up the call.

As we discussed above, all the call-identifying information of a managed VoIP call is available to the service provider who sets up and manages the VoIP call; it is fairly straightforward for the VoIP service provider to trace any VoIP calls made or received by its customers. In this article we only consider how the unmanaged VoIP calls could be effectively traced. Specifically, we consider using the VoIP flow itself, rather than the VoIP signaling, to uniquely identify the caller and callee of peer-to-peer Skype VoIP calls.

Tracing Model and Challenges

Generally, we want to link the caller and callee of potentially anonymized VoIP calls. Given any two different Skype peers A and B, we are interested in determining if A is talking (or has talked) to B via Skype peer-to-peer VoIP. Since we are only interested in finding out if some parties that we suspect have communicated via peer-to-peer VoIP anonymously, we only need to be able to monitor and intercept IP flows to and from those parties we suspect. In other words, we do not assume or require the global monitoring and intercepting capability.

The Skype peers could be behind firewalls and NAT, and the VoIP traffic between peers A and B could be routed through some low-latency anonymizing network. Since the Skype VoIP flows are encrypted from end to end, it is impossible to identify who the caller and callee are from the VoIP content. Because the Skype VoIP calls could pass through some low-latency anonymizing network, the original VoIP flow is broken into segments of VoIP flows, where each segment of VoIP flow has its own source and destination IP addresses. When there are multiple VoIP calls over the anonymizing network, it is difficult for the tracer to figure out which fragment of VoIP flow belongs to which VoIP call, even though he can intercept all the flows incoming or outgoing to the anonymizing network.

If the tracer could somehow identify some common characteristics from all the segments of one VoIP flow, and if those common characteristics are distinct enough for different VoIP flows, the tracer is able to correlate those VoIP flow segments and find out the IP addresses of the original caller and callee. Therefore, the key for tracking encrypted VoIP flows on the Internet is to identify some unique characteristics of the encrypted flow and determine the correlation among the VoIP flow segments.

A number of papers (e.g., [24–27]) have shown that the interpacket timing characteristics of interactive flows are distinct enough such that encrypted interactive flows could be effectively correlated (and differentiated) based on their interpacket timing characteristics.

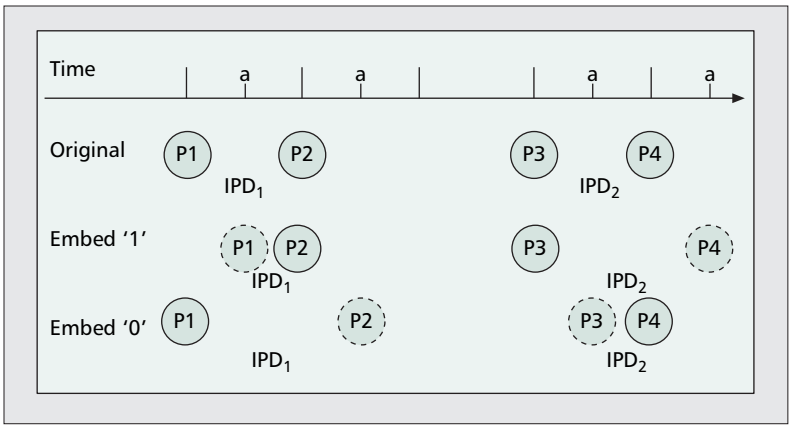
Unfortunately, the original interpacket timing characteristics of VoIP flows are not distinct enough. This is because the interpacket arrival time of VoIP flow is determined by the VoIP codec, and there are only a few commonly used codecs. Specifically, most VoIP flows have either 20 or 30 ms packetization interval. Therefore, all VoIP flows would have very similar interpacket timing characteristics, and passively comparing them would not be able to distinguish different VoIP flows.

In fact, when VoIP calls are

- Made from computer to computer
- Protected with strong end-to-end encryption
- Anonymized through low-latency anonymizing network

many people would intuitively think it is infeasible to track such VoIP calls on the Internet

To the best of our knowledge, there is only one published work [28] that addresses the problem of tracing anonymous VoIP calls on the Internet. As we show below, strong end-to-end encryption and existing low-latency anonymizing network does not necessarily provide the level of anonymity to VoIP calls that people would intuitively expect.



■ Figure 4. *Embedding a binary bit into interpacket delay.*

An Active Timing-Based Approach

In this subsection, we briefly describe our active-timing-based approach that can be used for effective tracking of anonymous P2P VoIP calls on the Internet.

Unlike all the other timing-based approaches, our approach is active in that it deliberately yet subtly makes the interpacket timing characteristics of the VoIP flows more unique. This is achieved by embedding a unique watermark into the interpacket timing domain of the upstream VoIP flows. Then the corresponding downstream flows can be effectively identified by checking if they have the embedded watermark or not.

In theory, the embedded watermark could be removed from the watermarked VoIP flow if the interpacket timing of the watermarked VoIP flow could be somehow restored to exactly 20 or 30 ms (depending on the codec being used). In practice, however, it is difficult to do so on those P2P VoIP flows. This is because

- The peer computers usually do not run hard real-time OS and do not have a precise timer
- There is no centralized gateway along the path of P2P VoIP calls
- It is infeasible for those anonymizing systems to determine if any incoming encrypted flow is VoIP or not

Therefore, it is practically infeasible to remove the watermark embedded in the P2P VoIP flows.

Suppose we have randomly chosen four packets P_1, P_2, P_3, and P_4 from a particular VoIP flow (shown in Fig. 4), whose arrival time stamps are t_1, t_2, t_3, and t_4, respectively. If we group these four packets into two pairs ($P1$, P_2) and (P_3, P_4), then we can obtain the interpacket delays (IPD) of these two pairs, denoted as IPD_1 and IPD_2.

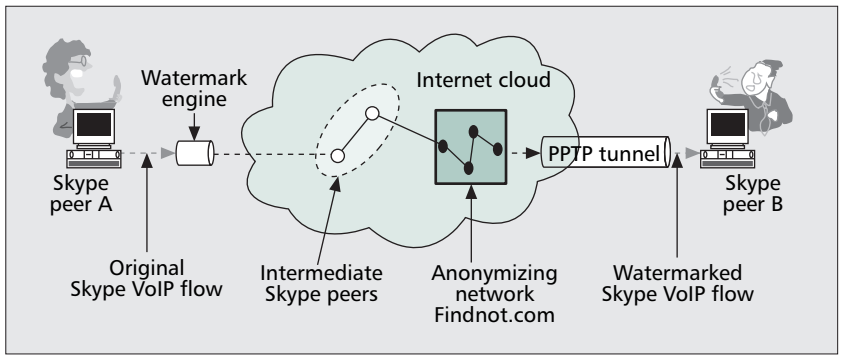
We can further calculate the normalized difference between the two IPDs: $IPDD = (IPD_2 - IPD_1)/2$. Of course, *IPDD* could be positive or negative. Because P_1, P_2, P_3, and P_4 are selected randomly, the distribution of *IPDD* is symmetric and centered around 0. To embed a binary bit '1', we deliberately delay the departure time of packets P_1 and P_4 for a period of time a. This would effectively shift the distribution of the original *IPDD* to the right by amount a, which means *IPDD* becomes more likely to be positive than to be negative. Similarly, we can embed binary bit '0' by shifting the distribution of original *IPDD* to the left by amount a. This can be achieved by deliberately delaying the departure time of packets P_2 and P_3 for a period of time a. To decode the embedded watermark bit, we simply use the same randomly selected packets and compute the corresponding *IPDD*. If the *IPDD* is less than or equal to 0, we would get decoded bit '0'; if the *IPDD* is greater than 0, we would get decoded '1'.

It is possible that the *IPDD* from randomly selected packets happens to be less than $-a$ (or greater than a). If we want to embed bit '1' (or bit '0'), we will get a decode error, since shifting *IPDD* to the right (or the left) by amount a does not make *IPDD* greater than 0 (or less than or equal to 0). While we can never eliminate this possibility, we can reduce such probability by utilizing redundancy techniques. Specifically, we can use more packets to get r *IPDD* ($r > 1$) and calculate the average ($\overline{IPDD}$) of these *IPDD*. Compared with *IPDD*, $\overline{IPDD}$ has smaller variance and is more clustered around 0. In fact, the error decoding probability (the probability that $\overline{IPDD}$ falls outside range $[-a, a]$) can be decreased to arbitrarily close to 0 with a large enough redundancy number r.

Another source of error comes from the active timing perturbation caused by the adversary who deliberately perturbs the interpacket timing in an attempt to corrupt the embedded watermark, or from the jitter introduced by the network naturally. Again, the negative impact of random timing perturbation (or network jitter) can be minimized by using the redundancy technique. It was formally proved in [27] that the above watermark scheme can achieve, with an arbitrarily small timing adjustment a, arbitrarily close to a 100 percent correlation true-positive rate and arbitrarily close to a 0 percent correlation false-positive rate at the same time against arbitrarily large (but bounded) timing perturbation of arbitrary distribution, as long as there are enough packets in the flow.

We have empirically validated the active-timing-based approach with real-time Skype VoIP calls over the commercially deployed anonymizing system findnot.com. Figure 5 shows the experimental setup. We used two computers as Skype peers A and B, respectively. Skype peer B was connected to some entry point of findnot.com via the Point-to-Point Tunnel Protocol (PPTP) [29] so that all its Internet traffic was forwarded and thus anonymized by the anonymizing network of findnot.com. As a result, Skype peer B's IP address was hidden from all others, and some exit point of findnot.com functioned as a proxy for Skype peer B.

We made 100 Skype VoIP calls between peers A and B, and we embedded 100 different 24 bit watermarks into the VoIP flows from Skype peer A to Skype peer B. The results showed that the encrypted and anonymized Skype VoIP flow could be made highly unique with only 3 ms timing adjustments on selected packets. With 1200 packets randomly



■ Figure 5. *Experimental setup for the real-time tracking of VoIP calls across the Internet.*

selected from VoIP calls of 90 s long, we were able to achieve a 99 percent true-positive rate and a 0 percent false-positive rate at the same time. These results demonstrate that it is feasible to track P2P VoIP calls on the Internet even if they are anonymized by the low-latency anonymizing network and encrypted from end-to-end.

Conclusions

In this article, we have examined the anonymity aspect of VoIP calls and have shown how the use of VoIP has substantially shifted the previous balance between privacy and security that exists in traditional PSTN calls. While VoIP makes it much easier for end users to achieve confidentiality and anonymity, it has also introduced significant new challenges for law enforcement agencies (LEAs) to conduct lawful electronic surveillance. We discussed approaches that can be used to make VoIP calls anonymous, and reported the latest research results on how anonymous peer-to-peer (P2P) VoIP calls can be tracked on the Internet. We showed that the use of strong encryption and existing low-latency anonymizing network at the same time does not necessarily provide the level of anonymity to VoIP calls that people would intuitively expect.

References

[1] VOIP/IP telephony statistics, http://www.techweb.com/wire/8707174
[2] Network Working Group, The Secure Real-Time Transport Protocol, IETF RFC 3711, Mar. 2004.
[3] P. Loshin, *Big Book of IPsec RFCS: Internet Security Architecture*, Nov. 1999.
[4] Network Working Group, Point-to-Point Tunneling Protocol (pptp), IETF RFC 2661, Aug. 1999.
[5] Network Working Group, Traditional IP Network Address Translator, IETF RFC 3022, Jan. 2001.
[6] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous Connections and Onion Routing," *IEEE JSAC* Special Issue on Copyright and Privacy Protection, 1998.
[7] D. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," *Proc. 13th USENIX Sec. Symp.*, Aug. 2000.
[8] Freedom Internet security, http://www.freedom.net
[9] M. Freedman and R. Morris, "Tarzan: A Peer-to-Peer Anonymizing Network Layer," *Proc. 9th ACM Conf. Comp. and Commun. Sec.*, 2002, pp. 193–206.
[10] "Feds: VOIP a Potential Haven for Terrorists, http://news.com.com/Feds+VoIP+a+potential+haven+for+terrorists/2100-1028_35236233.html?tag=nl
[11] FBI, letter to FCC, http://www.askcalea.com/docs/20040128.jper.letter.pdf
[12] C. Gulcu and G. Tsudic, "Mixing E-mail with Babel," *Proc. Network and Distrib. Sec. Symp.*, Feb. 1996, pp. 2–16.
[13] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a Type Anonymous Remailer Protocol," *Proc. IEEE Symp. Sec. and Privacy*, May 2003, pp. 2–15.
[14] The Anonymizer, http://www.anonymizer.com
[15] Findnot, http://www.findnot.com
[16] The free network project, http://freenetproject.org/
[17] The Entropy project, http://entropy.stop1984.com/en/home.html
[18] Gnunet, http://www.gnu.org/software/gnunet/
[19] Winny, http://www.geocities.co.jp/SiliconValley/2949/
[20] ITU-T Rec. H.323v.4, "Packet-Based Multimedia Communications Systems," Nov. 2000.
[21] M. Arango *et al.*, "Media Gateway Control Protocol (MGCP) v. 1.0," Network WG, RFC 2705, Oct. 1999.
[22] H. Schulzrinne and J. Rosenberg, "A Comparison of SIP and h.323 for Internet Telephony," *Proc. Int'l. Wksp. Network and Op. Sys. Support for Digital Audio and Video*, Cambridge, U. K., July 1998, pp. 83–86.
[23] J. Rosenberg *et al.*, "SIP: Session Initiation Protocol," IETF RFC 3261, June 2002.
[24] Y. Zhang and V. Paxson, "Detecting Stepping Stones," *Proc. 9th USENIX Sec. Symp.*, USENIX, 2000, pp. 171–84.
[25] K. Yoda and H. Etoh, "Finding a Connection Chain for Tracing Intruders," *Proc. 6th ESORICS*, 2000.
[26] D. Reeves X. Wang and S. Wu, "Inter-Packet Delay Based Correlation for Tracing Encrypted Connections through Stepping Stones," *Proc. 7th Euro. Symp. Research in Comp. Sec.*, Oct. 2002, LNCS-2502, pp. 244–63.
[27] U. Shankar*et al.*, "Multiscale Stepping Stone Detection: Detecting Pairs of Jittered Interactive Streams by Exploiting Maximum Tolerable Delay," *Proc. 5th Int'l. Symp. Recent Advances in Intrusion Detection,* Oct. 2002, LNCS-2516, pp. 17–35.
[28] X. Wang, S. Chen, and S. Jajodia, "Tracking Anonymous Peer-to-Peer VOIP Calls on the Internet," *Proc. 12th ACM Conf. Comp. and Commun. Sec.*, Nov. 2005, pp. 81–91.
[29] Network Working Group, "Layer Two Tunneling Protocol," IETF RFC 2637, July 1999.

Biographies

SHIPING CHEN (schen3@gmu.edu) received both his Master's degree in management science and engineering and his Bachelor's degree in information science from the University of Science and Technology of China. He is a Ph.D. candidate in information technology and a research assistant in the Center for Secure Information Systems at George Mason University, Fairfax, Virginia. His research interests include network security and privacy, VoIP security, and database security.

XINYUAN WANG (xwangc@gmu.edu) received his Ph.D. in computer science from North Carolina State University in 2004. He is currently an assistant professor in the Department of Information and Software Engineering at George Mason University. His research interests have focused on network security in general and intrusion tracing in particular. His other research interests include intrusion detection and response, viruses and worms, information hiding, steganography, and privacy and anonymity and their interactions with security. He is the inventor of the active watermark tracing approaches and has successfully applied active watermarking approaches in developing highly effective attack attribution systems.

SUSHIL JAJODIA (jajodia@gmu.edu) is BDM International Professor of Information Technology and director of the Center for Secure Information Systems at George Mason Universit. His research interests include information security, temporal databases, and replicated databases. He has authored five books, edited 24 books and conference proceedings, and published more than 300 technical papers in refereed journals and conference proceedings. He has served in different capacities for various journals and conferences. He is the founding editor-in-chief of the *Journal of Computer Security* and on the editorial boards of *IEE Proceedings on Information Security*, *International Journal of Cooperative Information Systems*, and *International Journal of Information and Computer Security*. His web page is at http://csis.gmu.edu/faculty/jajodia.html