EFFECTIVE IDENTITIES FOR TRUSTED INTERACTIONS IN CONVERGED TELECOMMUNICATION SYSTEMS

A Thesis Proposal
Presented to
The Academic Faculty

by

Vijay A. Balasubramaniyan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Computing

Georgia Institute of Technology
August 2011

EFFECTIVE IDENTITIES FOR TRUSTED INTERACTIONS IN CONVERGED TELECOMMUNICATION SYSTEMS

Approved by:

Professor Mustaque Ahamad, Advisor
College of Computing
Georgia Institute of Technology

Professor Wenke Lee
College of Computing
Georgia Institute of Technology

Professor Patrick Traynor
College of Computing
Georgia Institute of Technology

Professor Douglas Blough
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Arup Acharya
Services
IBM Research T.J. Watson

Date Approved: 30 June 2011

To my parents,

T.S. Balasubramaniyan and Mangala Balasubramaniyan,

for entreating me to study when I wanted to play . . . all the time.

ACKNOWLEDGEMENTS

I first walked into Georgia Tech, a Master's student, a little less than six years ago. As true for most international students my biggest concern was getting an assistantship. Most faculty, aware of this kept their doors closed except for Mustaque Ahamad, who was most welcoming. I walked into his office and after listening to an oral resume he gave me a set of papers to read and asked me to think about possible open ended challenges in securing voice over IP (VoIP) networks. Soon after, he took a chance on funding me and what followed were a series of enlightening conversations where Mustaque would never suggest what I should do or what path I take. He would only help me come up with the right questions and find my own answers. As a result I feel a a strong sense of ownership for the research presented in this thesis. For this and a whole lot more, I thank him.

The following summer I did the first of many internships at IBM Research, TJ Watson. What keeps bringing me back there is the fun I have working with Arup Acharya and the fact that I am so close to New York City. Arup introduced me to industry grade, feature rich, converged telecommunication systems where architecture diagrams do not fit a page and acronyms have acronyms. For great research summers and pocket money for the rest of the year, I thank him.

Towards the end of my PhD, I started working with Patrick Traynor. His enthusiasm and ability to lead by example pushed me into exploring areas that were previously alien to me. For making the last couple of years truly enjoyable and eye opening, I thank him. I would also like to thank the rest of my thesis committee, Doug Blough and Wenke Lee. During the initial years, discussions with Doug helped me become more methodical towards my research. Wenke has been a great yardstick for

judging the research merit of an idea. He has always asked the hard questions. I would also like to thank the rest of the GTISC family, Mary Claire Thompson, Alfreda Barrow and Lerverne Davis for accommodating the crazy administrative requests, my research collaborators, Lei Kong, Italo Dacosta, Frank Park, Younho Lee, Naveen Tamilmani, Aamir Poonawalla, Mike Hunter, Viswanathan Mahalingam, Arjun Maheswaran, Arunabh Verma and my office mates, Kapil Singh, Frank Park, Chaitrali Amrutkar, Saurabh Chackradeo and Hank Carter, for the fruitful discussions and the reading group sessions.

Almost every December I made it a point to go back home to my wonderful parents. Four weeks with Amma's food and Appa's conversations would do wonders for both the body and the mind. Fittingly, my first research idea took seed while I was back home. Even during paper deadlines my parents always provided great company when I would walk back home during the wee hours of the morning. For shorter breaks, there was the great option of visiting my two brothers, Karthik and Jai. Through this PhD, Karthik has become my closest friend. My summers at IBM research would see me spend the week in New York and the weekends with Karthik, his wonderful wife Sheela and my two sweet nephews, Sanjit and Abhinit, watching movies and playing cricket. Karthik's place was my proverbial home away from home. Without the help of my brother Jai, I would never have come up with as detailed an architecture for the VoIP lab, largely responsible in getting that first research assistantship position at Tech. On winters that I did not go to India, it was always a blast to visit him and his family.

At Tech itself I have had a great support system of friends. My girlfriend, Karishma Babu, has been by my side through the cycle of paper rejections and acceptances. I have not, like Rudyard Kipling philosophized, treated them equally, and Karishma will effusively stand testament to that fact. For the movies, board games and the so many ways she has helped, I cannot thank her enough. I would like to

thank my other friends, the fab four, Arpan Ghosh, Nivedhya Ramaswamy, Nithya Sambasivan and Ranjith Subramanian, for making the first two years here memorable with helping me pick a great car, and for the parties and the road trips galore, Vishakha Gupta for being a great friend, organizing all the social events that kept me sane and the timely dinner invites during hungry paper deadlines, my two wonderful roommates Ryan Maladen for the food binges at Indian restaurants and Romain Cledat for the great desserts and racquetball games, the 508ers, Renuka Apte, Neha Deodhar, Meeta Bajpai and Priyanka Tembey for the tea sessions and the awesome spring break vacations, the new kids on the block, Adit Ranadive, Smita Vaidya, Mukil Kesavan, Danesh Irani and Tushar Kumar.

Finally thanks also go to old friends who know how much this PhD means to me, Ishwar Sundaraman, for ensuring we always got in a game of cricket before I left India, Divya Gupta, for patiently listening to my many ramblings, Sandhya Menon, for making New York city as much fun as it is, and Akshay Rao, Arun Raghavan, Ganesh Valiappan, Anshul Sheopuri and Karthik Tamilmani for always being there.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Telecommunication systems have evolved significantly since their inception and the recent convergence of telephony infrastructure allows users to communicate through a variety of ways including landlines, mobile phones and Voice over IP (VoIP) phones. While cellular and public switched telephone (PSTN) networks use Caller ID to identify users, VoIP networks employ user ids, similar to email, to identify users. However, in all these networks this identity is locally asserted and is therefore easily manipulated. It is easiest to assert any identity within IP networks and this has resulted in VoIP spam (e.g., the recent Skype Computer Repair spam calls). As IP networks converge with other PSTN and cellular networks, it has also become easy to assert any Caller ID across these networks. The larger issue of Caller-ID spoofing has increasingly contributed to credit card fraud and identity theft. To address this, we introduce the notion of effective identity which is a combination of mechanisms to (1) establish identity of the caller that is harder to manipulate, and (2) provide additional information about the caller when necessary.

In this dissertation, we first look at the specific issue of determining the legitimacy (additional information) of a user id within IP networks to address the VoIP spam problem. We propose CallRank, a novel mechanism built around call duration and social network linkages to differentiate between a legitimate user and a spammer. We realize that any system that determines the legitimacy of users based on their social network linkages leaks private information. To address this, we create a token/credential framework that allows a user to prove the existence of a social network path between him/her and the user he/she is trying to initiate contact with, without actually revealing the path. We combine the privacy properties of two techniques in

cryptography: Delegatable Anonymous Credentials (DAC) and E-Cash to create this framework. We then look at the broader issue of determining identity across the entire telecommunication landscape to address the issue of Caller ID spoofing. Towards this, we develop PinDr0p, a technique to determine the provenance of a call - the source and the path taken by a call. In particular, we show that the codec transformations applied by multiple intermediary networks, in combination with packet loss and noise characteristics, allow us to develop profiles for various call sources based solely on features extracted from the received audio. In the absence of any verifiable metadata, these profiles offer a means of developing specific fingerprints that help uniquely identify a call source. We show that the audio can also provide valuable additional information. We use anomalies in timbre created by different undersea telecommunication cables to develop London Calling, a mechanism to identify geography of a caller. Together, the contributions made in this dissertation create effective identities that can help address the new threats in a converged telecommunication infrastructure.

CHAPTER I

INTRODUCTION

Telecommunications has evolved significantly since its inception in the 1800s to a thriving $4 trillion sector in 2010. The current telecommunication infrastructure allows users to communicate using a variety of technologies. Circuit switched landlines, which operate on Public Switched Telephone Networks (PSTN), continue to provide telephony to the majority of homes and businesses. Mobile phones now offer service to more than four billion users [161]. Voice over IP (VoIP) allows users to inexpensively communicate with each other irrespective of the geographical distances, with systems such as Skype [27] currently serving over 400 million users [25].

One fundamental question in a telecommunication system is when a person receives a call, should he answer it. Two aspects governing this decision are: (1) the identity of the caller, and (2) associated information about the caller. If a call recipient knows the caller, then it is easy for him to determine whether or not to take a call. Unfortunately, in telecommunication networks identity has always been locally asserted. In VoIP, user ids are self picked. In PSTN and cellular networks, identity is provided by Caller ID which is volunteered by the calling side. Further complicating this situation is that people often receive calls from people they do not know and yet it is important for them to answer that call. For example, consider a call from a friend of one's parent who is visiting the city and needs someone to show him around. In such cases, identity is not sufficient and the recipient needs additional information about the source of a call. We define effective identity to be a combination of provided credentials and inferred feature values about a particular caller that helps a call recipient determine if that call will result in a desirable interaction.

The lack of such effective identities in telecommunication networks has made it vulnerable to easy attacks. Within VoIP systems, since user ids are self picked, call spam attacks have emerged where attackers have created accounts claiming to be well established computer repair shops. The introduction of VoIP has also eroded much of the trust associated with traditional telephony, making it easy to claim any Caller-ID, resulting in Caller ID spoofing attacks. Caller ID spoofing has contributed significantly to credit card fraud, identity theft and disruption of 911 services. For example, in 2009, a single criminal ring used Caller-ID spoofing to steal close to 15 million dollars.

1.1 Dissertation Contribution

Creating effective identities in telecommunication networks has several challenges. First, from a call recipient's perspective, we need to identify which calls require only identity and which need effective identities to ensure that the call results in a desirable interaction. Second, we need to identify what additional information is useful in making effective identities. Third, since call interactions are extremely personal, we need to ensure that the additional information we provide does not reveal confidential information about calls to people not participating in a call. Fourth, we need to ensure that effective identity itself is robustly determined and cannot be easily manipulated. Finally, we need to measure our ability to reduce current attacks such as VoIP spam and Caller ID spoofing that exist in telecommunication systems due to a lack of effective identities. We hypothesize that **privacy preserving effective identities can be created in a converged telecommunication infrastructure to reduce VoIP spam and Caller ID spoofing**. This dissertation investigates mechanisms to create such effective identities.

We first start by looking at VoIP systems, where there have been many mechanisms for establishing basic identity. We provide details of these mechanisms in the

background and related work in Chapter 2. As mentioned before, there are many situations where people receive calls for the first time, from people they do not know and would be willing to answer it (e.g. parent's friend). This is known as the *introduction problem* in peer to peer systems. As current systems do not provide any additional information, attackers have exploited this to spam users into accepting calls by claiming to be friends and then going on to sell them unwanted products. In Chapter 3, we introduce CallRank, a system that provides both local and global reputation information about callers that can be used to differentiate between a legitimate user and a spammer. Our approach is motivated by the simple observation that a legitimate user typically makes and receives calls and many of the calls last for reasonable durations. On the other hand, a spammers/telemarketers goal is to deliver information to as many people as possible by making a large number of relatively brief calls. For a spammer, the call pattern is largely unidirectional with short call duration while it is bidirectional for legitimate users with relatively longer call durations. We take advantage of this difference in call patterns and create credentials that callers can provide to recipients as proof of an implicit level of trust. These credentials essentially determine Social Network (SN) linkages [35] between users, enabling us to distinguish between legitimate users and spammers. We also use call duration along with the Eigentrust algorithm [93] to develop a global view of the reputation of all users who either belong to or interact with a domain. We implement CallRank and demonstrate its ability to identify spammers with high specificity and sensitivity even in the presence of a significant number of spammers.

We realize that any system that determines the legitimacy of users based on their social network linkages leaks private information. To illustrate, let us suppose that Alice wants to prove to Bob that she is a legitimate user (and not malicious) by showing that that they have a good mutual friend in Charlie. To prove Charlie is a friend, Alice will need to reveal previous interactions with Charlie that indicate that

the two are friends. The more recent and the longer these interactions are, the more convinced Bob is of Alice and Charlie's friendship. However, revelation of these interactions is a sacrifice of both Alice's and Charlie's privacy. To address this, in Chapter 4, we introduce Privacy Preserving Grapevines, a token/credential framework that allows a user to prove the existence of a social network path between him/her and the user he/she is trying to initiate contact with, without actually revealing the path. We combine the privacy properties of two techniques in cryptography: Delegatable Anonymous Credentials (DAC) [36] and E-Cash [47] to create this framework. We show that though this framework has cryptographic overheads that affect call setup times, we can achieve practical tradeoffs to keep this call setup time low. In addition, this framework maintains the high specificity and sensitivity of CallRank.

We then look at the broader issue of determining identity across the entire telecommunication landscape to address the issue of Caller ID spoofing. Towards this in Chapter 5, we develop PinDr0p[1], an infrastructure to assist users in determining the provenance of a call - the source and the path taken by a call. Through a combination of signal processing and machine learning techniques, we show that regardless of the claimed source, the audio delivered to the receiver exhibits measurable features of the networks through which the call was delivered. For example, calls that traverse a VoIP network experience packet loss that results in perceivable effects in the final call audio. Such artifacts are noticeably absent in calls that have only traversed cellular or Public Switched Telephone Networks (PSTNs). In particular, the codec transformations applied by multiple intermediary PSTNs, VoIP and cellular networks, in combination with packet loss and noise characteristics, allow us to develop profiles for various call sources based solely on features extracted from the received audio. In

[1]Our mechanisms take advantage of audio and path artifacts that, like the sound made by the drop of a pin, are largely unobservable to the human ear.

the absence of any verifiable metadata, these profiles offer a means of developing specific fingerprints that help identify a particular call source. Using these fingerprints we show that we are able to distinguish between calls made using specific PSTN, cellular, Vonage, Skype and other hard and soft phones from locations across the world with high accuracy. In Chapter 6 we extend techniques developed in Pindr0p to use anomalies in timbre created by different undersea telecommunication cables to develop London Calling, a mechanism to identify geography of a caller.

Our results provide strong evidence to support our hypothesis that it is possible to create privacy preserving effective identities that reduce VoIP spam and Caller ID spoofing. We finally conclude this thesis in Chapter 7.

CHAPTER II

BACKGROUND AND RELATED WORK

2.1 Background

Telephony networks are exceedingly complex systems. While once designed, manufactured and run by a single company, today's networks are an elaborate combination of many different technologies. We offer a very high-level description of these systems, how voice is encoded in them and the transformations that occur as voice crosses between different classes of networks.

As shown in Figure 1, there are three general classes of telephony networks. PSTNs represent traditional circuit-switched telephony systems. These networks are generally characterized by lossless connections and high fidelity audio. While pieces of the core of some of these networks are being replaced by IP connections, these provider owned links are tightly controlled to ensure near zero packet loss. Like PSTN systems, cellular networks have a circuit switched core, with portions currently being replaced by IP links. While these networks can have considerably different technologies deployed in their wireless interfaces, their cores are extremely similar. Finally, VoIP networks by name run on top of IP links and generally share the same paths as all other Internet-based traffic. Accordingly, VoIP systems virtually always experience packet loss.

In all these networks there are two parts to enable calling, (1) signaling that establishes and tears down the call, and (2) media which carries the voices of the call participants. These are achieved by different mechanisms in each of these networks and we discuss these mechanisms in the next couple of subsections.

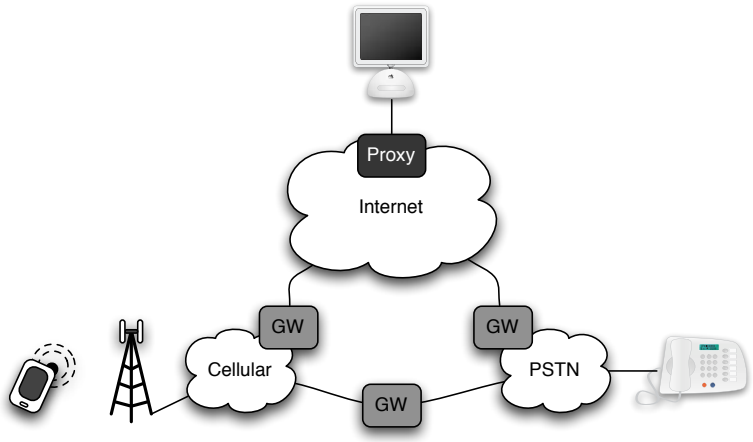


Figure 1: A high-level description of modern telephony systems. Note that a call between two endpoints may cross a variety of networks. At each gateway, calls are re-encoded using that network's codec.

2.1.1 Signaling

The core signaling mechanism used for call setup, routing and control in PSTN and cellular networks is the common channel signaling system no. 7, SS7. Within the SS7 protocol stack, the Integrated Services Digital Network (ISDN) User Part (ISUP) defines the procedures to setup, manage and release trunk circuits that carry voice and data calls. Despite its name, ISUP is used for both ISDN and non-ISDN calls. To initiate a call, the calling party goes off hook and dials the directory number of the called party. These numbers are transmitted as DTMF digits to the closest telephone exchange's service switching point (SSP). SSPs are switches that originate or terminate calls. The SSP then transmits an ISUP Initial Address Message (IAM) to the destination SSP. This IAM consists among other things the dialed digits and the voice trunk circuit reserved for this call. The calling party name (Caller ID) is also transmitted as an optional parameter. When Caller ID is requested to be blocked, this information is not sent as part of the IAM. The IAM is routed via a packet switch called a signal transfer point (STP). An STP routes each incoming message to an outgoing signaling link based on routing information contained in the SS7 message.

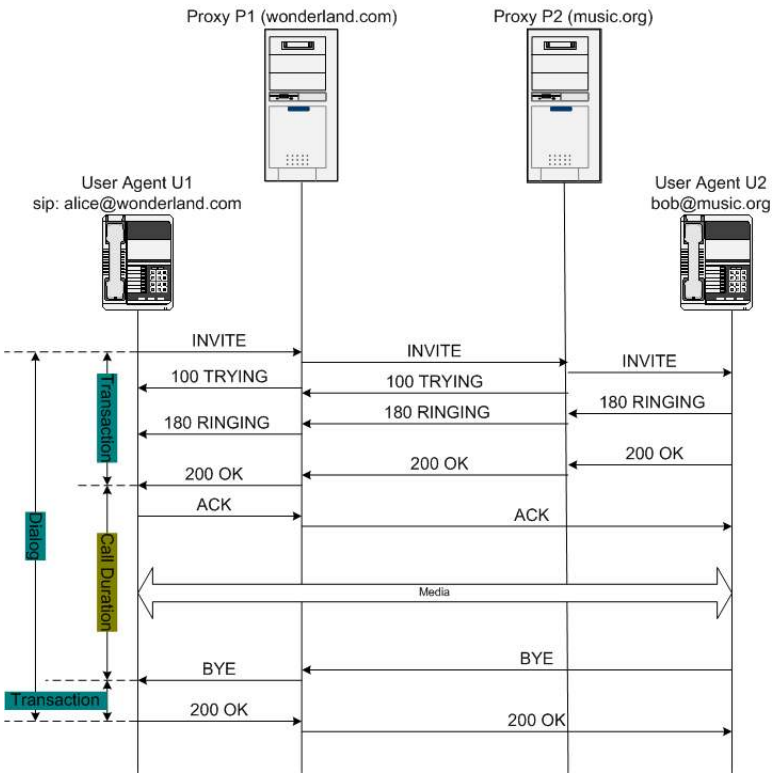


Figure 2: SIP Call Trapezoid. Call duration represents the time between the end of call setup (200 OK) to the start of call teardown (BYE)

Once the destination SSP confirms that the called party's line is available for ringing, it transmits an acknowledgment which translates to the ring tone heard by the calling party. Once the called party picks up a phone an ISUP answer message (ANM) is transmitted to the calling party and the two parties then use the reserved trunk to transmit voice between them.

The signaling mechanism for VoIP is similar to SS7 and is enabled using either the Session Initiation Protocol (SIP)[126], proposed by the IETF or H.323[155] proposed by the ITU. Since SIP is de facto standard, we discuss how call setup and teardown is achieved using SIP.

For two users to communicate with each other using SIP, they need to know each other's SIP URIs (Universal Resource Identifier). SIP then uses an application overlay consisting of proxy servers and location services to locate these end points.

Table 1: Audio Codecs and their typical deployment.

Codec	Networks	Applications
G.711	PSTN, VoIP	Standard Telephony
GSM-FR	Cellular	Cellular Telephony
iLBC	VoIP	VoIP over Cable
Speex	VoIP	XBox Live
G.729	VoIP	SkypeOut/SkypeIn

A typical SIP call trapezoid is shown in Figure 2. When *Alice* identified by SIP URI *sip:alice@wonderland.com*, calls *Bob*, *sip:bob@music.org*, the call request message ($INVITE$) is sent to the proxy server responsible for the *wonderland.com* domain, $P1$. $P1$ then determines how to route the call to the proxy responsible for *Bob*'s domain, *music.org*, $P2$. Once $P2$ receives the request it looks up user *Bob* and then routes it to the appropriate endpoint. On receipt of the $INVITE$ message, *Bob*'s user agent (UA) starts to ring, shown by the 180 *Ringing* in Figure 2. When *Bob* picks up the phone, the UA sends a 200 OK message. This initial message exchange forms the call setup transaction. When *Bob* or *Alice* hang up, the respective UA sends a BYE message and this initiates the call tear-down transaction. Call duration represents the time between the end of call setup (200 OK) to the start of call teardown (BYE) (see Figure 2). Call duration is the basic building block of the CallRank scheme proposed in Chapter 3.

2.1.2 Media

Voice is encoded and decoded in each of these networks using a variety of *codecs*. Specific codecs are selected for different networks based on competing goals including sound quality, robustness to noise and bandwidth requirements. While a large number of codecs exist, we describe and study the five most commonly used narrow band codecs in this work. We summarize these codecs and their typical environments in Table 2.1.2.

The codec used all over the world in PSTNs is G.711 [149], with North America

and Japan using the *mu*-law compression algorithm and Europe and the rest of the world using A-law. Both the algorithms generate a 64 kbps (20 ms audio frames) Constant Bit Rate (CBR) stream for speech sampled at 8kHz, which is relatively bandwidth intensive when compared to other codecs. In cellular networks, the GSM full rate (GSM-FR) [80] codec was the first digital cellular coding standard and is still widely used in networks around the world. Unlike G.711, which is a waveform coder, GSM-FR uses predictive coding, which is more common among modern codecs and allows a large reduction in bandwidth requirements, with GSM-FR having an average bit rate of 13 kbps.

A plethora of codecs have been specifically designed for VoIP systems. The Internet Low Bit-rate codec (iLBC) [75] is extremely robust to packet losses and operates on a bit rate of 13.33 kbps (30 ms audio frames) and 15.20 kbps (20 ms audio frames). iLBC is a mandatory standard for VoIP over Cable and is also used by Google Voice and Skype [27]. Speex [22] is a Variable Bit Rate (VBR) codec that supports a wide range of bit-rates from 2.15 kbps to 44 kbps and uses 20 ms audio frames. Speex, in addition to being supported on many VoIP soft phones, is commonly used in gaming teleconferencing systems such as Xbox Live [23]. A large number of VoIP systems also use G.729 (10 ms audio frames) [150], which requires very low bandwidth as it supports a CBR of 8kbps. Skype also uses G.729 when making and receiving calls to landlines and mobile phones (SkypeOut/SkypeIn service). It is also used by most Cisco hard IP phones [26]. Finally, a number of VoIP phones also support G.711, which is used in PSTN systems.

Audio must be reencoded when passing between two different telephony networks. For instance, whereas the audio in a call between two PSTN users is likely to only have been encoded in G.711, both G.711 and GSM-FR will be applied to the audio for a conversation between users on a PSTN and cellular network, respectively. Encoding changes occur in media gateways located at the edge of telephony networks, meaning

that VoIP calls can traverse multiple Internet autonomous systems without necessarily being reencoded. Through this infrastructure, phone calls can be delivered seamlessly between users. *It is these transformations and the characteristics of the underlying networks that we seek to measure to establish call provenance proposed in Chapter 5.*

2.2 Related Work

The lack of common signaling mechanisms between the different networks has resulted in easy assertion of any identity. This has resulted in both VoIP spam and in Caller ID spoofing. We first look at related work in the VoIP spam area and show how are solution compares with the others. As a large number of proposed solutions including ours use social networks to identify the legitimacy of a user there are immediate privacy risks when such information is exchanged. This tradeoff between accountability and privacy has been studied in peer to peer networks that exchange token information to determine the legitimacy of a peer and we highlight this research. Finally, as we broaden our context across the entire telecommunication landscape, we look at how other systems have addressed the provenance of information and discuss how that relates to determining the provenance of a call.

2.2.1 VoIP Spam

Rosenberg et al. [128] provide a comprehensive reference for the various possible solutions that can be explored for VoIP spam. Techniques from email spam such as Blacklists, Statistical Blacklists, Greylists, Whitelists and Consent Based Systems are adopted for VoIP in [57, 135, 82]. The techniques mentioned above are subverted easily by the creation of new identities, a mechanism used in attacks such as the Sybil attack[64]. We show that CallRank, however, is resistant to these kind of attacks in Section 3.3.3.1. Spam based on anomalous characteristics of a spam call is explored in [145], [134] and [163]. However, the characteristics being monitored are easily subvertible, once known. Strong authentication is probably the best counter measure

against SPIT, however techniques based on DKIM[83], P-Asserted-Identity[89] and SAML[156] specified in [145] and [128] will only be as successful as the fraction of user base that utilizes it. Establishing absolute identity on the Internet is always going to be a hard problem. It is unlikely that we will have a practical and a universally deployable solution based completely on absolute identity.

In the democratic setting of the Internet, reputation based techniques seem to be most practical and effective. Dantu et al.[57] and Rebahi et al.[120] suggest the use of buddylists and user ratings for buddies to create dynamic localized whitelists. However, this restricts the group of users that can call to strictly the user's SN linkage and it requires explicit user feedback in the form of ratings. CallRank on the other hand, uses call duration, which is recorded automatically by the system without requiring explicit user action.

2.2.2 Accountability and Privacy

Accountability and fairness in P2P systems have predominantly used tokens[103, 55, 28]. Anagnostakis et. al.[28] advocate the notion of transferable tokens and show the improvements in scalability and redundancy afforded by introducing such tokens. An alternative to tokens for accountability in P2P systems is the use of micropayments[174, 159, 87, 84]. In essence, all these schemes prove the existence of a transfer path by revealing information about that path and therefore have significant privacy concerns.

Adding privacy requirements to incentive mechanisms(like tokens) has been studied extensively in reputation and recommender systems[119] and social networks[50], utilizing a host of cryptographic techniques. Laurent et. al.[46] use group signatures, while Carminati et. al.[50] use digital signatures to provide anonymity. Kai et. al.[167] use group signatures to add anonymity to the micropayment scheme proposed in [174]. For our setting, we have demonstrated how the underlying techniques used

by these schemes still leak privacy. Belenkiy et. al.[37] try to achieve accountability without losing privacy by using an E-Cash mechanism to provide a currency model in P2P. However their system does not support transferable coins. A transferable E-Cash mechanism using the meta proof technique is described in Canard et. al.[49]. However, the meta proof technique is a general circuit based proof that is inefficient in practice. To solve this, we create Privacy Preserving Grapevines, where we allow users to act as banks in their own right, creating, issuing and transferring tokens to each other, since tokens are meaningful only to the issuer (he is the one getting spammed).

2.2.3 Information Provenance

The concept of data provenance in computing was first studied in database systems. The proposed techniques seek to identify the source of a piece of data and the process by which it arrived at the database [45, 79, 38]. Such information can be proactively added at the source and transformation points as metadata [70, 59] or reactively obtained through techniques such as query inversion [169, 56]. Such techniques have been adapted and extended to other platforms including web servers with trusted hardware [111]. The presence of such mechanisms provides a significantly improved infrastructure for performing audits and determining data quality [109].

More recently, a number of researchers have attempted to provide provenance information for networks. Traceback techniques [131, 176, 81] attempt to determine the true path of packets in the presence of potentially spoofed source information. Such information can either be added directly to the packets as metadata [131, 140, 172, 118], or by state stored and queried from within the routers themselves [177]. A range of watermarking tools also exist to identify the provenance of flows in IP networks [165, 99, 85]. The diversity of telephony networks (i.e., circuit switched PSTN, cellular and VoIP) makes such watermarks extremely difficult. Specifically,

metadata introduced in one network (e.g., watermarks, path information) is generally lost when the call is transmitted over another network.

We are not aware of previous work that attempts to identify the provenance of a phone call in a diverse telephony environment. However, techniques in a purely Internet-based environment have been considered [146]. Perhaps the closest to our work are caller identification (Caller-ID) services that provide the caller's number or name in PSTN and mobile networks. Calls originating from IP networks traditionally have no unique associated number or name and therefore cannot be used to identify the caller [139]. Moreover, a variety of techniques already exist to spoof phone numbers [12]. Artifacts of calls themselves may provide significant provenance information. Specifically, because call quality relies greatly on a combination of the codec [106, 21], the range of end devices [44] and network degradations [61, 125, 102], the detection of these characteristics using tools designed to measure single-ended call quality [61, 106, 125] can potentially be used to further improve the provenance of a call.

In this thesis we will explore several related problems that arise due to a lack of effective identities. We start by discussing CallRank and how it addresses the VoIP spam problem in the next chapter. In chapter 4, we propose an extension to CallRank that continues to use social network linkage information to differentiate legitimate users and spammers while addressing the privacy risk of sharing such information. We finally discuss establishing the provenance of calls in Chapters 5 and 6 and show how that addresses the problem of detecting fraudulent calls and Caller ID spoofing in a diverse telephony infrastructure.

CHAPTER III

CALLRANK: COMBATING VOIP SPAM

Voice over Internet Protocol (VoIP) systems rely on an IP network to set up voice calls and transmit voice packets. The growing popularity of VoIP, the relatively low cost of access to IP networks, and the vulnerabilities that exist in systems connected to such networks makes VoIP an attractive tool for spammers. Spammers and telemarketers will use VoIP to make unsolicited calls and to send voice mails for the same purposes for which email spam is currently used. VoIP spam would not only degrade our confidence in telephony but it would be more difficult to handle because of the real-time processing requirements of voice calls. Examples of large scale VoIP spam already exist - a company sent out voice mails to all its customers detailing its initial public offering[132]. If we are not able to combat VoIP spam effectively, we face an unhappy future where picking up a ringing phone would be a frustrating experience and voice mailboxes would become clogged with advertisements for unwanted products.

The first stage of voice communication is call setup, a handshake mechanism between the caller and the call recipient after which the phones start ringing. At this stage the only information provided is the identity of the caller and the call recipient. It is only after the call recipient accepts the call, that voice media is exchanged. A spam engine that filters based on the media content, however successful it is, will not be able to prevent the phone from ringing constantly. In addition unlike email, voice packets must be delivered to the user synchronously. Any delay in delivery due to spam engine processing will result in degraded call quality. Thus, an effective method for dealing with VoIP spam must rely on a robust identity of the caller rather than call content. However, determining the exact identity of a user on the Internet is a

hard problem. It is sufficient if we are able to differentiate between a legitimate caller and a spammer. In this work, our focus is on developing a scheme that achieves this goal.

This work proposes CallRank, a novel mechanism built around call duration, to differentiate between a legitimate user and a spammer. Our approach is motivated by the simple observation that a legitimate user typically makes and receives calls and many of the calls last for long durations. On the other hand a spammer's/telemarketer's goal is to deliver information to as many people possible, in as little time, by making a large number of short calls. A spammer will typically receive no calls or a much smaller number of calls. The difference in call patterns is that, for a spammer, the call pattern is largely unidirectional while it is bidirectional for legitimate users. We take advantage of this difference in call patterns and use call duration to create *call credentials* that callers can provide to call recipients.

The following simple scenario shows how our call credential based approach can be used to identify spammers. Assume that *Alice* makes a call to *Bob*. If *Bob* picks up the phone and talks to *Alice*, after completion of the call, a call credential can be generated signifying that *Bob* and *Alice* trust each other enough to talk for a certain duration of time. The longer the call duration, stronger is the call credential. As basic intuition suggests, if a user receives calls of significant duration on a regular basis, it is likely that he/she is a legitimate user and not a spammer. There are several ways in which call credentials can be created when calls are made. For example, when *Alice* calls *Bob* and talks to him for t time, she can create a call credential and provide it to *Bob* who can use the credential when making another call to show that he is not a spammer. It is also possible that the recipient of the call (*Bob*) generates a call credential for *Alice*. Although several of these options exist, in this work we explore a mechanism where a caller, when he/she speaks to a call recipient, provides a call credential to the call recipient.

For each user we use call credentials to determine Social Network (SN)[35] linkages. We also use call duration along with the Eigentrust algorithm[93] to develop a global view of the reputation of all users that either belong to or interact with a domain. For a spammer to be successful in the resulting system, CallRank, he/she must get other legitimate users to call and speak to him/her for significantly large durations. We believe this will be extremely hard as people rarely call up a spammer. If they inadvertently do make a call to a spammer, the conversation will not last for very long.

The following are the key contributions of this work:

- We introduce call duration based credentials as the uniform underlying mechanism to support a number of techniques to determine if a caller is a spammer.
- We explore the use of SNs based on the call credentials to allow two users to make a call.
- If SN linkages are unavailable between users, we use a variation of the Eigentrust algorithm to assign global reputations based on call durations.
- We perform a detailed evaluation of CallRank and show that we are able to achieve low false negative and low false positive rates even in the presence of a significant fraction of spammers.

The rest of the chapter is as follows. Section 3.1 discusses SNs, and Section 3.2 discusses the Eigentrust algorithm. The key components of CallRank are presented in Section 3.3. An evaluation of CallRank and its results are discussed in Section 3.4.

3.1 Local Reputation Using Social Network Linkages

In CallRank, SNs are used to decide when to accept a call credential. SNs model associations that exist between a set of entities (typically humans). A distinctive feature of these networks is their tendency to cluster, measured by the clustering

coefficient[166]. Mathematically, an SN can be described as a graph $G = (V, E)$, where V, the set of vertices/nodes represent people and E, the set of edges represents some relationship/association between the people. G is referred to as the community. Consider a three vertex community consisting of nodes A, B and C. If a particular node, A, is connected to the other two nodes, B and C, then for the community to exhibit a high clustering coefficient B and C must also be connected. This tendency to form triangles from wedges is the nature of a highly clustered SN. In a voice communication system if there is a scenario where user A calls user B and user B calls user C, then due to the similar clustering nature in these systems, it is highly likely that user C will at some point call user A. This high likelihood coupled with call credentials is used in CallRank to provide a local mechanism to determine if a caller is a spammer or not.

3.2 Global Node Reputation Using Eigentrust

We utilize the Eigentrust algorithm[93] to determine the reputation of a set of peers based on their interactions. In Eigentrust, each peer i decides a normalized local trust value for another peer j, based on the number of satisfactory and unsatisfactory transactions it has had with that peer. This value is represented as c_{ij} It then uses a transitive notion of trust to aggregate these local trust values to a system wide reputation value for all peers. If $\overrightarrow{t}$ represents a vector containing these values, the eigentrust algorithm determines this vector by solving $\overrightarrow{t} = (C^T)^n * \overrightarrow{e}$ for $n = large\ number\ of\ iterations$. C is the matrix containing the normalized local trust values $[c_{ij}]$, $\forall i, j$. $\overrightarrow{e}$ is the unit 1-norm, that is $e_i = 1/m$, where m is the total number of peers in the system. $\overrightarrow{t}$ converges to the left principal eigenvector of C. In case there exists pre-trusted peers P, we need to ensure that these end up with high reputations. Therefore to converge faster, we can use $\overrightarrow{p}$, instead of $\overrightarrow{e}$ where $p_i = 1/|P|$ if $i \epsilon P$ and $p_i = 0$ otherwise. The system to solve, in the presence of

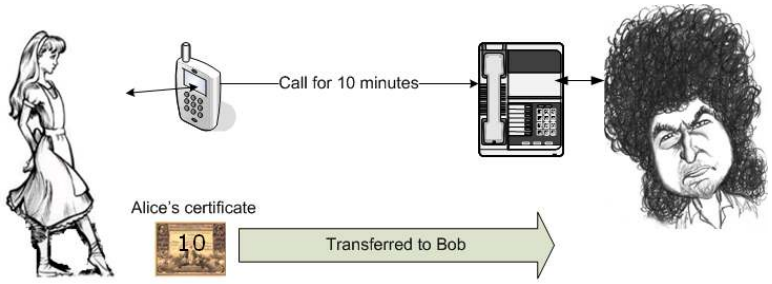


Figure 3: Call duration represented as a reputation credential. This is the building block for both establishing local reputation through social network linkages and global reputation through Eigentrust

pre-trusted peers, is $\overrightarrow{t} = (C^T)^n * \overrightarrow{p}$.

3.3 CallRank Overview

3.3.1 Voice Call Duration

Consider a call from Alice to Bob where the call duration is 10 minutes, as shown in Figure 3. This, to us, represents an implicit statement that Alice trusts Bob enough to speak to him for 10 minutes. On termination of the call, Alice's user agent (UA) will then automatically hand a secure call credential to Bob stating that "Alice spoke to Bob for 10 minutes", represented by CC_{AB}. We ensure its security through cryptographic primitives discussed in Section 3.3.5. The next section discusses how we can combine this credential and SN theory to determine what call credentials can be trusted.

3.3.2 Using SNs to Accept a Call Credential

Consider, once again, the system as described in Figure 3, following which Bob talks to Charlie for 15 minutes. At this point Bob's UA hands a credential capturing this information to Charlie, CC_{BC}. At a later point in time assume Charlie tries to call Alice. If Charlie's UA presents CC_{BC} to Alice's UA at call setup time, then Alice can accept the call since she knows Bob (as she has recorded information of the call from her to Bob). In a general scenario, the caller UA will present to the call recipient's

UA a set of credentials when initiating a call. The call recipient's UA will see if any of the credentials can be used to establish a SN linkage and then decide either to accept or reject the call. Such a decision may consider several factors to determine how important or useful a particular credential is. For example, when Alice receives call credential CC_{BC} from Charlie, which has been generated by Bob, the factors that will influence Alice's decision to accept the call are: (1) How strong is CC_{BC}?, and (2) How fresh is CC_{BC}?

The strength of the credential is dependent on the call duration value encapsulated within it. Thus, Bob speaking to Charlie for an hour will generate a stronger credential than Bob speaking to Charlie for a couple of minutes. Alice's UA also checks for the freshness of the credential. For this we assume that the UA's have access to approximately synchronized common clocks and we believe most phones will be time synchronized in a commercial VoIP deployment. Alice's UA can be configured with a policy stating that only call credentials with durations greater than a particular threshold, say T_{CD}, and timestamps within a certain time window shall be considered. We use the average call duration of the user as the value for T_{CD}, that is

$$T_{CD} = \frac{\sum Duration\,of\,Calls\,made\,by\,user}{Total\,number\,of\,calls\,made\,by\,user}.$$

A simpler scenario is when Alice speaks to Bob and Bob later wants to talk to Alice. Bob can use the credential that Alice provided to him. In this case there is a direct relationship between caller and call recipient and the call can be accepted. In general, calls are accepted only if there exists, between caller and call recipient either a direct relationship, or a transitive single hop SN linkage. We restrict the linkage to a single hop because then callers can only use credentials directly presented to them. This restricts misuse of credentials and keeps the design simple.

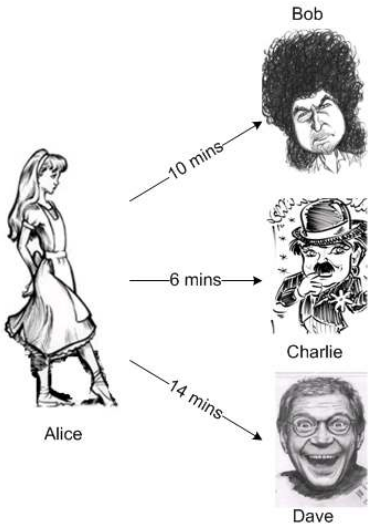
In our evaluation of CallRank, each UA maintains a record of all the people he/she called and a list of call credentials from users who made calls. The decision to accept or reject a call is then at the UA level and no other SIP component needs to get

involved. This forms a scalable load distributed solution as each UA is responsible for the calls it accepts or rejects. In most commercial phones, similar call history information is maintained under *Dialled Calls* and *Received Calls*. We can extend *Received Calls* to also store the call credentials.

3.3.2.1 Threats to SN Based Scheme

If a spammer needs to defeat our SN based model and make a call to a particular user, he/she will have to penetrate the immediate SN of the user. Consider the scenario where a spammer wants to call Alice. He/She will either have to get a call credential directly from Alice or from someone to whom Alice makes calls. Since it is unlikely that a legitimate user, such as Alice, or her immediate SN will call the spammer and talk to him/her for sufficiently long periods of time, the spammer will find it hard to obtain such a credential.

Assume the spammer manages to convince a user Bob (who is part of Alice's immediate SN) to talk to him/her for a sufficient duration. This may happen when Bob inadvertently calls the spammer once. Since the spammer now has a credential from Bob, he/she is able to spam everyone who makes calls to Bob including Alice. However, the freshness constraint of the credential will only allow the spammer a short time window where he/she can spam users who call Bob. If the spammer, on the other hand, is able to get Bob to call him/her regularly, then he/she will have a constant supply of fresh credentials. In such a case, Alice on being spammed, can now decide that she will no longer accept calls which present call credentials from Bob. Again, the spammer is only successful for a short duration. If the spammer needs to disseminate information to a large number of users, he/she will need to penetrate all their possibly disjoint SNs in a similar fashion. The down side of our SN scheme is that there will be situations where even legitimate users will not be able to use call credentials because there exists no SN linkage between them. The global reputation



	A	B	C	D
A	0	.33	.2	.47
B	.5	0	.5	0
C	.08	.35	0	.57
D	0	0	0	0

Figure 4: The left principal eignevector of the sample matrix is the global reputation of this matrix. The matrix cannot be rank deficient and therefore each row must at least have one non zero entry

scheme discussed in the next section will be used to address this problem.

3.3.3 Global Reputation Using Eigentrust

Over the course of some period of time, assume that Alice talks to Bob, Charlie and Dave and the talk times are as shown in Figure 4. We can use call duration to represent the reputation value that Alice implicitly assigns to people she calls. Formally, the normalized local reputation value provided by a user i who calls a user j is calculated as

$$r_{ij} = \frac{Duration\,of\,all\,calls\,to\,j}{\sum_k Duration\,of\,all\,calls\,to\,user\,k}. \quad (1)$$

This ensures that r_{ij} is between 0 and 1 and for any row i, $\sum_{\forall j} r_{ij} = 1$. This is analogous to the normalized local trust value in the Eigentrust system[93]. The advantage of normalizing is that reputation values are not arbitrarily high or low. This prevents users who form a malicious collective from assigning a high reputation value to other users in the collective and low value to legitimate users.

The first row in Figure 4 represents Alice's reputation values towards Bob, Charlie and Dave based on equation (1). Similarly the reputation values that Bob, Charlie

and Dave assign to each other and Alice can be calculated and form the subsequent rows in a reputation matrix, R. For the system comprising only of Alice, Bob, Charlie and Dave, the reputation matrix is shown in Figure 4. If we need a system wide view of reputation values then we have to aggregate these local reputation values. We discussed in Section 3.2 that this is the leading left eigenvector, λ of the matrix R. λ_i then represents the reputation of user i as perceived by the system as a whole. In calculating the leading eigenvector, we use the power method specified in [76]. We deviate from the method by normalizing the trust vector obtained at the end of each iteration using its 1-norm (the method in [76] uses the 2-norm). Using the 1-norm ensures that the final eignevector, λ, is such that $0 \leq \lambda_i \leq 1$, $\forall i$ and $\sum_i \lambda_i = 1$. Thus the system as a whole has a total possible reputation of 1 and each individual has some fraction of this reputation. Using a 1-norm over a 2-norm does not seem to affect the convergence rate in our experiments.

Proxies that provide billing services maintain call duration information for all users within their domain. The proxy is, therefore, the best place to maintain and update the reputation matrix. Periodically it can calculate and update the leading eigenvector of the matrix. In addition the proxy can also include users (from other domains) who have either made or received calls to or from this domain in its reputation matrix. In CallRank, when a proxy server receives a call request, it consults the eigenvector calculated to obtain the reputation value for the caller and appends this information to the request. The call recipient can then decide based on a threshold value if calls will be accepted or not. In our evaluation only the call recipient's proxy appends a reputation value which makes the value hard to be tampered with. We can also have the caller's proxy provide a reputation value but that will be less trustworthy.

3.3.3.1 Threats to Global Reputation Scheme

We discussed how it is hard for a spammer to penetrate a legitimate user's SN and thus compromise CallRank's effectiveness. It is equally hard for the spammer to obtain a high global reputation value. This is because the reputation value is based on call interactions with a number of users and takes into account the reputation of these users. If a spammer needs to have a high reputation value, he/she will need a significant number of moderately reputed users to call him/her and speak for sufficiently long durations. This is an unlikely occurrence. A legitimate user, on the other hand, will have a high reputation value due to call interactions with other legitimate users (a feedback loop). This implies even fairly sophisticated attacks like the Sybil attack[64] can be thwarted because coming to the system with a new identity implies no SN linkages or reputation and this is detrimental towards making calls.

3.3.4 The Introduction Problem

When a new legitimate user joins a VoIP system, he has no social network linkages in that system and a low reputation value. This will change if other users call him/her increasing his reputation value and providing him/her with call credentials. However, other users are unaware of his entry into the VoIP system. In order to notify other users he will need to make the first call. In CallRank, however, all calls he makes will be flagged as spam calls, which amounts to a false positive. We can fix this by combining CallRank with other schemes proposed for VoIP spam such as an audio Turing test or a computational puzzle. When a user is flagged as a spammer, he will then be subject to the Turing test or a computational puzzle or even a personalized question from the call recipient (what is my high school nickname). The call is accepted if the caller is able to successfully answer any of these tests. In our simulation we have not included such a Turing test and this forms part of our future work.

3.3.5 Call Credentials

The call credential needs to have accurate and secure information about the call durations. A call credential CC consists of A, the identity of the caller, B, the identity of the call recipient, t, the call duration and TS, the time stamp of the call along with a digital signature of the same information. We assume that each user has a public/private key pair which is used to generate the digital signature. If not already available, this pair can be generated by the UA on first use. Associating a public key with a particular user is done with key rings in the manner proposed in [100], thus avoiding the use of an infrastructure such as PKI.

The accuracy of the information within the credential can be verified by the proxy which also records call duration information. We assume the proxy has an accurate value of call duration as it provides billing services. Therefore, the proxy does not need call credentials for calculating reputation values. In fact, if the proxy is also used to determine the SN linkage for a call, we do not need call credentials. The call duration information recorded by the proxy is sufficient. However, we believe moving the SN linkage detection to the proxy makes the system unscalable.

To understand the call credential better, we consider what it means from a human perspective. This credential is a record of the user's past observed behavior in the system or his/her call history. If the user is an active member of a particular VoIP community, making and receiving calls, he/she will accumulate the community relevant credentials through his/her interactions, making it easier to identify him/her accurately within the community. This is exactly how it works in the real world. If for some reason there is a sufficiently long break from the community then when he/she re-enters, he/she will once again have to reestablish himself/herself. Since credential collection is transparent, users can use the system with minimal impact on usability. Using call duration as a building block has the following advantages. It is (1) implicit, (2) quantifiable (3) easily verifiable, and (4) easily understood.

3.3.6 Discusssion of CallRank Algorithm

To summarize, the CallRank algorithm works as follows. On receipt of a call setup message, the UA first checks to see if any call credentials presented by the caller belong to users to which the UA has made calls. If such a credential is found and it satisfies the policy duration and freshness constraints, the call is accepted. If no credential satisfies the constraints then the algorithm checks the reputation value of the caller. If this satisfies a particular acceptable reputation threshold, then the call is accepted. Otherwise, the call is deemed spam and is rejected. Another technique like a Turing test may be used at this point but the present implementation of CallRank does not support this.

CallRank does have some limitations. The first limitation is that legitimate users, who make a large number of outgoing calls but receive very few incoming ones, would not be able to collect call credentials. Typical examples are emergency services and banks. Since these systems are part of critical infrastructure, they can be seeded with high global reputation values. The second concern is one of privacy because the collection of call credentials provides user with call history information of their immediate SN. We discuss how we address this in the next chapter.

3.4 CallRank Evaluation

We simulate CallRank with a synthetic call workload to evaluate its effectiveness. In particluar, we measure how quickly users can distinguish between legitimate callers and spammers and the results are discussed in Sections 3.4.1, 3.4.2 and 3.4.3. We study legitimate caller acceptance in Section 3.4.4.

Our initial experimental setup consists of DNS, proxy and statistics servers and user agents. Initially, only the DNS and the statistics server are running. Each proxy server registers with the DNS server, and the user agents register with the proxy. User agents either behave as reputed users (seeded with high reputation values), legitimate

users (users who make legitimate calls but are not seeded with high reputation values), or as spammers. A legitimate or a reputed UA makes calls to other phones with inter call and call duration values that are Poisson distributed. The choice of call recepient is Zipfian distributed. Spamming UAs, however, make calls to as many other UAs as possible.

Call setup goes through proxies which consult the DNS server and then route the call to the proxy in the call recipient domain, which in turn forwards to the call recipient. During the learning period, which can be set, a call recipient will accept all calls. After the learning period, a call is accepted or rejected based on call credentials and reputation value. All call interactions are recorded at the statistics server which track number of accepted and rejected calls for both legitimate users and spammers. Our initial setup consists of three domains each served by a proxy server and 200 users initially registered in each domain. 1% of the 600 users are reputed. The number of spammers and regular users is varied based on the experiment. We use a simulated call workload model. To simulate call processing for a sufficient period of time, 100 seconds of system time models 1 day of simulated time.

3.4.1 Effect of Spammers

The first set of experiments determines the effect of spammers on CallRank. Three runs are conducted where the spammers present are varied from 1%, 10% and 20% and the fraction of spam calls accepted for each case is measured. The results are as shown in Figure 5 which plots the fraction of spam calls accepted with time. When legitimate users join the system, they have a learning period during which time they accept ALL calls. This period is essential for the user to gather credentials and build reputation. However, they are vulnerable to spam calls. The spammer thus needs to detect a new user within this learning period time window and then send all the spam they can generate. In our simulation the learning period for all UAs is fixed at 1 day.

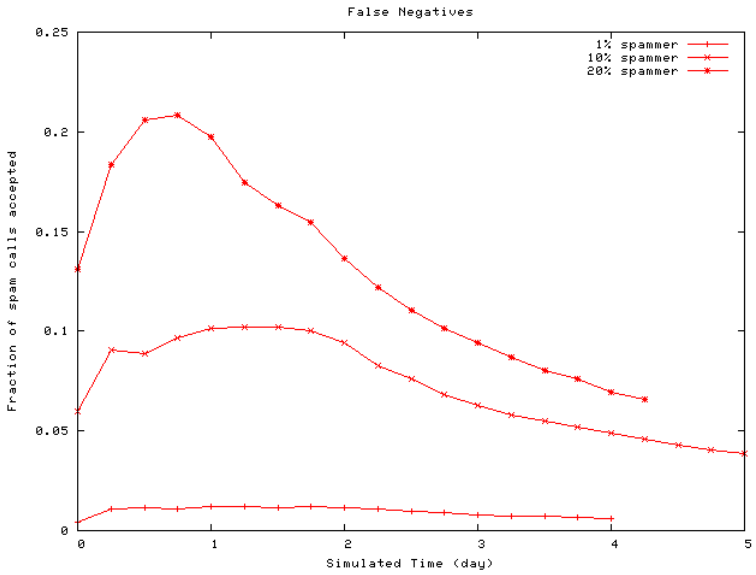


Figure 5: Effect of spammers. This experiment assumes an aggressive threat model where spammers identify new users in the system as soon as they join.

All three scenarios initially show increase as spammers learn about more and more legitimate users and are able to send spam to them successfully. This increase lasts roughly for the learning period and then starts decreasing rapidly. This is because legitimate users, using the CallRank scheme, are now able to differentiate between spammers and legitimate users. For all three scenarios there are no new spam calls accepted after 4.5 days.

As the percentage of spammers increases from 1% to 10% to 20% the probability of some spammer discovering a legitmate user increases and the ability to send larger amounts of spam increases as well. This is seen in Figure 5 as each of the curves shows higher false negative rates of 1% to 10% to 22% respectively. Thus, the false positive rate increases linearly with the number of spammers. However, these numbers are contingent on the fact that legitimate users are discovered by spammers within their short learning period time window. If the legitimate user is undiscovered then the rates will drop down even further. In fact, once a legitimate user crosses his/her learning period he/she is able to identify spammers (old and new) with ease.

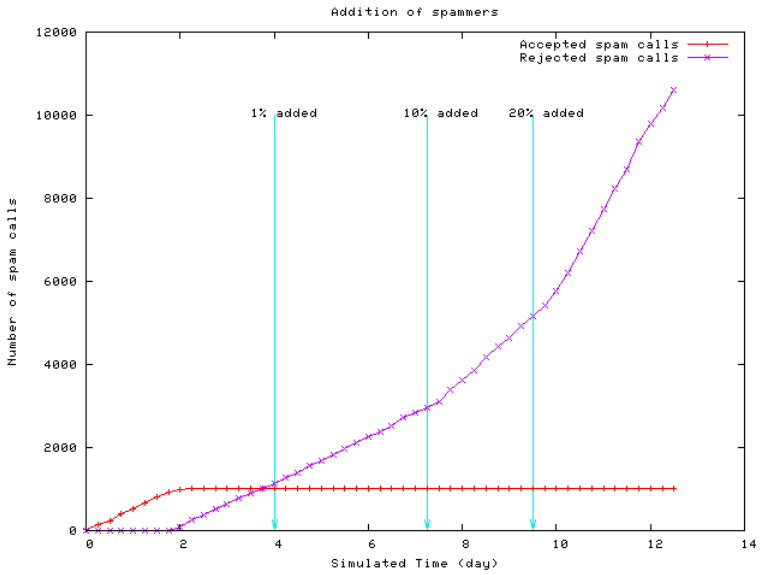


Figure 6: Impact of new spammers. As users stay in the system longer, CallRank helps them to be robust against new spammers.

3.4.2 Addition of New Spammers

We start with an initial population of 600 user agents, 1% of which are reputed UAs, 10% spammer UAs and the rest are legitimate UAs. We wait until the system stabilizes, that is no new spam calls are accepted or no new legitmate calls are rejected. From Figure 6 we see this occurs after 2 days and the number of accepted spam calls has saturated around 1000 calls. We then add spammers, 1%, 10% and then 20% of the current UA population. As seen the addition of these spammers does not increase the number of accepted spam calls illustrating that CallRank's mechanisms ensure that new spammers do not affect existing legitimate users. The reason behind this is that a new spammer, when introduced, does not have any SN linkage or reputation. Therefore, existing legitimate users will not accept any calls originating from them. Thereafter a spammer, due to his behavior, will not improve either his SN or reputation implying that at no stage will a legitimate user accept a call from him. This is a big advantage of the CallRank scheme where older users by virtue of their good call history become more adept at rejecting spam calls. Attacks such as

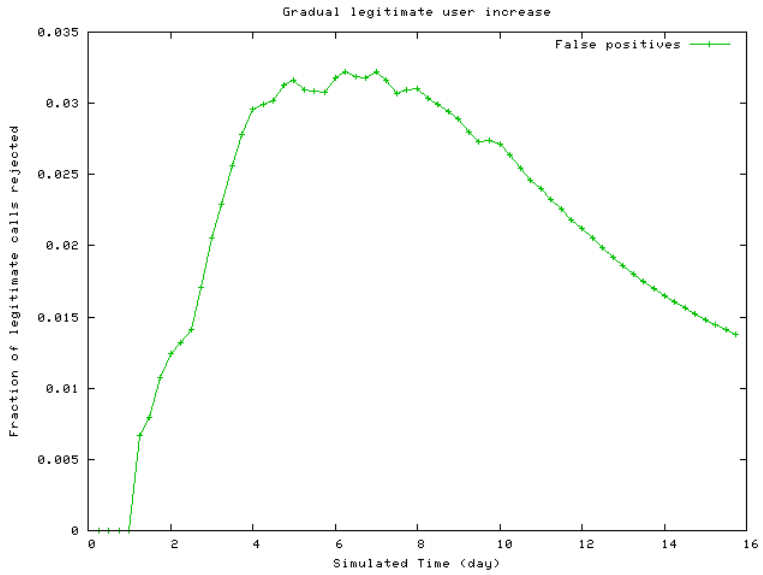


Figure 7: Adding legitimate users. The false positive rate increases to 3% but reduces significantly as legitimate users are recognized by the system to be legitimate

the Sybil attack which work very well against techniques such as blacklists will not succeed against CallRank.

The addition of new spammers generates more spam and we can see this in the increase in the number of rejected spam calls in Figure 6. At each stage of the introduction (marked by arrows) we can see an increase in the slope of rejected spam calls thus corroborating CallRank's effectiveness.

3.4.3 False Positives

Although not shown in the previous experiments, the false positive rates are also extremely low. For example, in the simulation run that involved 600 users, 1% of which are reputed and 10% spammers, there were only 3 calls that were wrongly rejected to give a false positive rate of .02%. This low rate is because all users are introduced at the same time and their learning periods coincide. Therefore, all users were simultaneously aware of the rest of the users by the end of this period. However, in a realistic scenario, users join a system over a period of time. To simulate this we created 600 users, 200 in each domain, over a period of 10 days. Within a domain

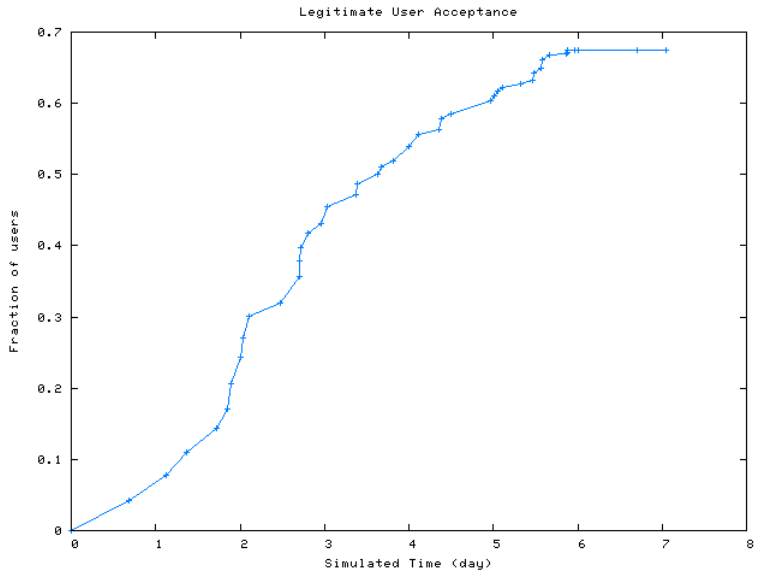


Figure 8: Legitimate user acceptance. If a user behaves legitimately, it takes him 3.5 days of simulation time to be accepted by half the user base.

users are created 3 hrs (simulated time) of each other. The false positive rate of such a system is as shown in Figure 7. The rate initially increases to a high of 3% and then reduces gradually. This is because, when a user joins the system, he has no SN linkage and no reputation which by CallRank's perspective is the characteristics of a spammer. Therefore, most of his calls will be rejected. However, if the user behaves legitimately, this rate drops soon enough showing that CallRank is able to determine that the user is legitimate. False positives do not have the same connotation as in the email world, where it implies a permanent loss of information. In the VoIP world, since interactions are synchronous, a user whose call is rejected can be asked to take an audio Turing test. This will only result in occasional longer call setup times, typically occurring when the user initially joins the system.

3.4.4 User Acceptance

We studied the acceptance of a legitimate user into a system containing 1000 existing users. This is shown in Figure 8. As we can see a legitimate user is accepted by half the total user base in 3.5 days. However, this factor can be used by a spammer to

alternate between being a legitimate user and a spammer, thus, providing him with the ability to spam a large set of users. This threat is not as large as it seems, because behaving as a legitimate user entails getting people with significant SN linkages or moderate reputation values to talk to the spammer on a continuous basis. When a legitimate user joins a voice communication system, almost immediately there are other legitimate users who talk to him thus creating his SN and establishing his reputation. This happens naturally for most people who have an established life outside the VoIP system. Their SN linkages or their reputations are just extensions of their real world persona. On the other hand a spammer has no existence outside the VoIP system, and so, legitimate users will never call him when he gets introduced into the system.

We also see that the graph in Figure 8 saturates at 70% (say set S) of the user base. That implies that anytime this user calls any of the users belonging to the remaining 30% (S'), he will be treated as a spammer. This is because there exists no SN linkage between the user and S' and the user's reputation value is significantly lower than users in S'. From our logs we see that S' consist of either the initially pre-reputed users or users that have been in the system in a legitimate fashion long enough to have become extremely reputed. This behavior is beneficial as it implies spamming users who are very reputed is going to be extremely hard.

3.5 Conclusion

Within VoIP systems there are multiple mechanisms to establish identity. However, for scenarios where users need additional information (e.g., when it is a parent's friend calling and the identity just says it is Joe calling) there are no suitable options to be able to take a call . To create these effective identities, we proposed CallRank, a system that uses call duration in conjunction with social network linkages and global reputation to determine if a user is a spammer or not. Our simulation explored the

effectiveness of CallRank and showed that it adapts over time, allowing users with legitimate call history to make calls easily while defeating spammers. In addition, our system is able to accept new legitimate users relatively easily while ensuring that new spammers are not able to affect existing users. This shows that CallRank is able to create effective identities that reduce VoIP spam. However in the next chapter, we discuss how expanding CallRank to consider social network paths of larger than two hops immediately reveals confidential information. To address this limitation we then develop a system that creates effective identities that are privacy preserving and continue to be robust against VoIP spam.

CHAPTER IV

PRIVACY PRESERVING GRAPEVINES: PRIVATELY CAPTURING SOCIAL NETWORK INTERACTIONS

As discussed in the previous chapter VoIP systems suffer from the spam problem [4, 16], primarily due to the inability of these systems to determine whether a user initiating contact for the first time is honest or malicious - the introduction problem. For example, in IM systems such as Yahoo Messenger or Google Talk, users explicitly invite people that they would like to chat with. To counteract this, in AIM[1], spammers provide unsolicited content as part of the initial invite request itself. Some systems like Google Talk allow users who have had prior email correspondence to automatically chat with each other. Automatic introduction is especially important in real time systems like VoIP, where a call needs to be accepted or rejected as soon as it is received. To illustrate, consider a scenario where Alice's father's friend's son, say Bob, would like to talk to her about admission to a university program (or job openings at her workplace). Social network (SN) theory suggests that higher the number of such weak ties between users, higher the likelihood of of a new direct tie being established between them[77, 33]. In VoIP, such a tie could be gleaned by looking at the call graph between users. In this case, there would exist a call path between Alice and Bob. Alice should not have to explicitly determine whether there exists such a call path as she could be subjected to spam in the process. We, thus, need an automated framework that is able to *establish the existence of a SN call path between two users that are trying to communicate for the first time.*

In addition to the introduction problem, it is equally important to determine whether a user would like to continue communicating with people that he has been

introduced to. To illustrate, consider a user who is travelling to France and calls a travel agent to make trip plans. If the travel agent does not specialize in flight tickets to France, he could get fellow travel agents (his SN) to talk to the user. As long as the travel agent and/or his SN provides valuable information to the user, there is continued communication between them. However, when the travel agent starts contacting the user with promotional offers, the user will stop taking his calls. Since this information is unsolicited by the user, it again constitutes spam. Therefore, in addition, to being able to determine SN call paths, the framework should be able to *capture the willingness of a user to continue communicating with a particular user.*

In essence, to provide a good user experience in the presence of a spammer threat model, a system needs to address two different challenges. The first allows users without a direct link to communicates with each other, and the second monitors the quality and validity of a link that exists between two users. CallRank[29], tries to address this problem by encapsulating call duration as a digitally signed call credential that is transferred from a caller to call recipient. The call recipient uses this call credential to talk to the user or to the user's immediate friends. Since a spammer hardly receives calls and when he does, finds it hard to engage users in conversation, he is unable to obtain call credentials necessary to call and spam legitimate users. Specifically, in CallRank, at the end of a VoIP call between Alice(caller) and Bob(call recipient) that lasts 10 minutes, Alice issues a digitally signed call token to Bob, represented by $T^{A \to B}$. At a later instance when Bob wants to talk to Alice's friend, Charlie, with whom he has had no previous direct interaction, he presents $T^{A \to B}$ to prove to him that someone in his SN (namely Alice) was willing to talk to him. The factors that influence Charlie's decision to accept this call credential include how well he knows Alice, how long was the call and how recent was it. This provides Charlie great control on the calls he accepts. However, he also gets to know precisely when

and for how long Alice and Bob talked, something that violates their privacy. CallRank restricts itself to immediate friends (two hops) as the loss in privacy, illustrated above, is aggravated as the number of hops increase. Though we used the notation CC_{AB} in CallRank, we use $T^{A \to B}$ in this chapter as we consider a larger number of hops and need to clearly establish what constitutes a SN path. The two notations are equivalent. The CallRank setting clearly shows how SN call history provides a valuable mechanism to differentiate between regular users and spammers. However, these credentials are not privacy preserving. In addition, since they only allow a two hop SN, they are restrictive and lose valuable weak tie information that can be obtained by considering a larger hop SN.

In this chapter, we create a token framework that uses delegatable anonymous credentials (DACs)[36] to create N hop transferable tokens that allow a user to prove the existence of a transfer path between him and the user he is trying to initiate contact with, without actually revealing the path. If a token transfer is associated with a VoIP call then the token transfer path represents a chain of calls between two callers. This information can be used by legitimate users to prove the existence of a weak social tie (father's friend's son) between them and the user they are trying to call. In addition, we need these tokens to be single use to capture a user's continued endorsement of a direct link(strong social tie). Towards this we extend DACs with techniques from E-Cash[47] to create single use tokens with the ability to identify token double spenders. Single use tokens also ensure that malicious users cannot indefinitely reuse tokens that they either obtain directly or through some call path. We implement the token framework using the Pairing Based Cryptography (PBC) library[24] and utilize it in the VoIP setting to explore its performance in the presence of a spammer threat model. We believe other communication systems and SN based services can also use this framework with minor modifications.

This chapter makes the following contributions:

1. We identify the requirements for a framework that allows a new user, Bob, to prove the existence of SN call path between him and Alice, without revealing the actual path. In addition, the framework allows us to capture Alice's willingness to continue communicating with Bob.

2. We create a transferable single use token mechanism that extends delegatable anonymous credentials[36] with techniques from E-Cash[47] to realize this framework.

3. We provide an implementation of this framework using the PBC library and experimentally evaluate the costs associated with its operations.

4. We apply this framework to a VoIP setting and demonstrate that it can combat the spam problem with low false positive and false negative rates.

The rest of the chapter is organized as follows. In section 4.1 we discuss the requirements of the desired framework, followed by possible approaches in section 4.2. We show that none of these approaches satisfy all the requirements and we develop our solution by first discussing the building blocks: DACs and E-Cash in section 4.3. In section 4.4 we discuss how to combine DACs and E-Cash to create our single use privacy preserving transferable token framework. We discuss implementation details, and results that include operation times of our framework and the performance of the framework with respect to the VoIP spam threat model in section 4.6.

4.1 Token Setting and Required Properties

An example multi-hop call chain is shown in figure 9 and provides the setting for our token framework. In this setting, a user A_1 calls another user A_2, speaks for a certain duration and at the end of the call issues a token, T^{A_1} to A_2. A_2 can use this token to call A_1 back at a later time. In this example, A_2 subsequently calls A_3 and at the end of their call transfers T^{A_1} to A_3. A_2 could also issue his own token and we discuss

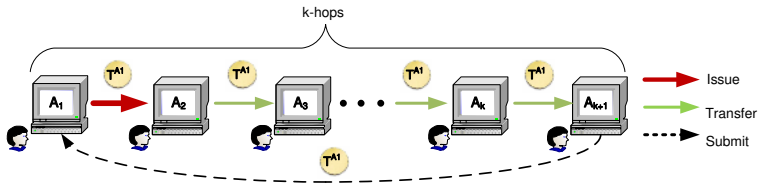


Figure 9: Multi-hop Token Transfer. A token from A_1 is transferred k hops until it is finally used by A_{k+1} to talk to A_1. The token must be constructed to prove that this $k-$hop path is legitimate.

a good strategy of deciding when A_2 issues his own token or transfers someone else's token in section 4.6.2. For now, A_3 gets the transferred token and can, once again, either use this to talk to A_1 or transfer it further. As shown in figure 9 this token is subsequently transferred to a user A_{k+1}, k hops away. Finally, A_{k+1} decides to use the token to talk to A_1 with whom he has had no previous interaction and presents T^{A_1}. Using T^{A_1}, A_1 can decide whether to accept or reject the call. In fact, since any user $A_i, 2 \leq i \leq k+1$ can use the token to talk to A_1, at the very least, the token should contain the identity of the token issuer. We assume this information does not need to be anonymized and argue that requiring this is both inefficient and does not provide a greater level of privacy.

Considering this setting, the three broad goals for a token are:

1. It can prove the existence of a transfer path between two users trying to establish communication for the first time. When A_{k+1} calls A_1, the token should convince A_1 that there exists a transfer path between them.

2. It can capture the willingness of a user to continue communication with a particular user and his SN. Tokens given to A_2 and his SN should not allow unlimited accessibility to A_1 as such a scheme can be misused.

3. It should achieve 1 and 2 above in a privacy preserving and efficient manner.

These goals translate to one or more of the properties listed below:

Unforgeability and Verifiability: The unforgeability property requires unforgeability of the token with respect to its issuing entity and the transfer path information that it carries. Specifically, when user A_1 calls user A_2, the token issued at the end of the call, T^{A_1}, should be unforgeable. This implies no other user should be able to issue a token on A_1's behalf. As the token is transferred, information about each transfer is appended to the token and this information should also be unforgeable. Specifically, when the token claims that it was transferred from user A_{i-1} to A_i then such a transfer should have actually occurred. This ensures that when A_1 finally receives the token from A_{k+1}, the unforgeability of the token issuer and the unforgeability of the transfer path information will allow A_{k+1} to prove the existence of such a path. We make the assumption that honest users transfer tokens only during calls and we note that without an all observing trusted third party there is no way of ensuring token transfers are tied to a call. Malicious entities may choose to transfer the tokens without a call. Despite this, a malicious entity should not be able to issue a token on behalf of an honest user or claim the existence of a transfer or a transfer path without it having occurred. The verifiability property requires that along with the token issuer, any user should be able to verify that the token is indeed issued by the issuer and the transfer path information is verifiably correct. This prevents the existence of bogus tokens in the system that are only discarded when they are finally submitted to the issuer.

A scheme that satisfies the above requirement can address goal (1). On the other hand, a user's willingness (or unwillingness) to continue interactions with another user, is useful in capturing the evolving nature of interactions. Since connections can be fleeting, as in the travel agent example, or can go away after a longer association (for example, relationships gone sour), deactivation is necessary. Also, a malicious entity might gain the trust of users and then start behaving maliciously. Essentially,

a scheme that assumes a user's behavior is going to always remain the same fails to realize any of these scenarios. In order to address goal (2) the token needs to satisfy the following requirement:

Single Use Tokens: Revisiting the travel agent example, we see that an infinitely reusable/non revocable token cannot capture the user's unwillingness to talk to the travel agent and his SN. Therefore tokens need to either be single use, have a restricted lifetime or support a revocation policy. Single use tokens provide a fair exchange for a users' interaction time. In the VoIP setting, if A_1 talked to A_2 for 10 minutes then the token T^{A_1} provides A_2 or his social network the ability to talk to A_1 for a proportional period of time. If A_1 is no longer willing to talk to A_2, as in the travel agent example, then A_2 only has a fixed supply of $A_1's$ tokens that he will eventually run out of. A more time sensitive approach is the use of token lifetimes where tokens expire after a specified time limit. However, determining what is a good token lifetime is hard, particularly when tokens are transferable, as the time between token issue and token use will vary. In addition, token lifetimes reveal information about the time of token issue. A more elaborate mechanism is incorporating token revocation. Anytime an issuer would like to deactivate a link to another user and his SN, he could send a token revocation to the user. However, this would require the user to keep state of all the other users to whom that token was transferred. All users who received this token would also need to maintain similar state and an elaborate revocation propagation mechanism would need to be put in place. In this light, single use tokens seem a practical token control mechanism to gauge a user's willingness to interact. They also allow issuers to decide the number of tokens to issue, thus requiring malicious users to obtain a steady supply of tokens for any user they wish to spam. However, single use tokens do not prevent token double spenders, users who transfer the same token to different users. Since only one of these tokens will be honored by the issuer, an honest user's call might be rejected in the process. Therefore, in addition to being

single use, it is important to be able to identify a token double spender.

Privacy: For honest users, a token is transferred during a call and contains sensitive call information. In achieving the above two requirements the scheme should ensure that details of a call should only be known to users directly participating in that call. Consider a call sequence: $A_{i-1} \rightarrow A_i \rightarrow A_{i+1}$ for token T^{A_1}. In this case when A_i transfers the token to A_{i+1}, A_{i+1} should only be able to identify that the token was issued originally by A_1 and that it has been verifiably transferred at each hop culminating at A_i. The token should not reveal the identities of previous holders of the token, including the fact that it was transferred from A_{i-1} to A_i. Therefore, a user in the transfer chain should only know who the token issuer is, who the token was received from and to whom it is being transferred. A user not in the transfer chain(example, someone who snoops a token off the wire) can at most know the identity of the token issuer as this information does not reveal any of his interactions. This notion of privacy should be preserved for all the contents of the token. We assume that the token issuer, for tokens issued by him, never acts maliciously.

Efficiency: The token scheme will essentially need to support the following operations: (a) token issue, (b) token transfer, (c) token submit and (d) double spender identification. In relation to VoIP, a caller at the beginning of a call submits a token. The call recipient accepts the call if the token is correct and is not duplicated, else the call is rejected. At the end of an accepted call the caller either issues a token of his own or transfers another users' token. Since these operations are tied to call setup and teardown in VoIP, they must be efficient in practice.

4.2 Possible Approaches

Before arriving at our proposed solution, we considered a number of possible approaches. None of them satisfy all the requirements but provide insights into the

challenges for creating a feasible scheme. We provide a brief overview of these approaches.

The simplest construct would be creating a token using a message authentication code (MAC) under the secret key of the issuer. The token issuer can always verify the validity of the token once it is submitted back to him. However, none of the other users can verify that the MAC was indeed generated by the issuer. To address this, we can create tokens using digital signatures (DS)[50]. The token issuer signs the token with his secret key and any user can verify that the token is generated by the issuer. However, this does not verifiably prove the existence of a transfer path. In addition, a malicious entity can snoop the token off the wire and make many copies of the token and transfer it across different paths. Even if there exists some serial number mechanism that prevents the token from being reused, there will be no way of identifying the user who made copies (double spent) of the token. Since the token double spender cannot be caught, tokens themselves become useless and cannot really ensure fair use of the system.

To prove the existence of a transfer path, user certificates could be employed to validate the transfer. If a user, A_1 wants to issue a token T^{A_1} to user A_2, he can associate a certificate with the token by signing A_2's public key with his secret key, $Cert^{A_1}(pk^{A_2})$. A_2 can use his secret key to prove to any user that he holds a valid certificate from A_1. A_2 can transfer the token to A_3 and in doing so, needs to provide a similar certificate for A_3, $Cert^{A_2}(pk^{A_3})$. A_3 now holds the token and the associated certificate chain $(Cert^{A_1}(pk^{A_2}), Cert^{A_2}(pk^{A_3}))$ to prove that he is the valid owner of the token. To prove the validity of a token any user must show an associated certificate chain that leads up to him. However, this clearly reveals all the interactions that have occurred so far. For example, when user A_3 further transfers the token to A_4, he has to reveal the certificate to prove token validity, which in turn reveals the interaction $A_2 \rightarrow A_3$. To hide the identity of a user in a certificate chain,

we can assume each user belongs to a group and use group signatures. Since we don't have to hide the token issuer's identity, A_1 can issue a certificate signing A_2's group public key with his secret key, yielding $Cert^{A_1}(pk^{A_2^G})$, where A_2 belongs to group A_2^G. When A_2 transfers the token to A_3, he can prove that he is part of group A_2^G by using the group secret key. He then provides a certificate of the form $Cert^{A_2^G}(pk^{A_3^G})$ to complete the token transfer. The certificate chain $(Cert^{A_1}(pk^{A_2^G}), Cert^{A_2^G}(pk^{A_3^G}))$ allows A_3, who is part of the group A_3^G to prove he has a valid token. When A_3 wants to further transfer the token to A_4 he can reveal this certificate chain. A_4 only gets to know that the original issuer of the token is A_1, and that some member of group A_2^G transferred the token to A_3. He no longer gets to know the identity of A_2. This scheme seems to capture the transfer path in a privacy preserving manner except for one problem. A_4 may transfer the token back to A_2 as he does not know that A_2 was previously an owner of this token. Though the associated credential chain is of the form $(Cert^{A_1}(pk^{A_2^G}), Cert^{A_2^G}(pk^{A_3^G}), Cert^{A_3^G}(pk^{A_4^G}))$, due to the uniqueness of the embedded information in the token(for example, the serial number), A_2 knows this is the same token that he transferred to A_3. Due to the deterministic nature of the way the token grows in size (this cannot be avoided[52]), A_2 also knows that this token has undergone only one transfer and therefore knows $A_3 \rightarrow A_4$, again a loss in privacy. In addition, for group signatures, clients have the overhead of creating sub groups and electing group managers. We could use ring signatures but since members of a ring need not voluntarily participate, the trustworthiness of a transfer path significantly degrades. We could avoid using groups or rings completely by using a zero knowledge (ZK) proof system to hide the identity of previous owners of a token. However, just like the group signature scheme, when a previous owner of a token sees the token again, he will be able to glean private interaction information. Detecting a cycle is impossible as a privacy preserving solution cannot reveal previous owners of a token. However, if cycle detection is impossible then we need to limit the maximum number

Table 2: Scheme Comparison

Scheme	**Unforgeability Verifiability**	**Single Use**	**Privacy**	**Efficiency**
MAC/DS	*X*	*X*	✓	✓
Certificate	✓	*X*	*X*	✓
Group Sign.	✓	*X*	*X*	✓
E-Cash	✓	✓	✓	*X*
DAC	✓	*X*	✓	✓
Our scheme	✓	✓	✓	✓

of hops that a token can be transferred. An added requirement to our token scheme is that, in order to avoid looping in a cycle forever, *the scheme should be able to restrict the number of hops.*

The occurrence of cycles in a token transfer path forms the hardest challenge in determining a scheme that satisfies our goals. For a token to not leak privacy, we observe that *all the information it carries must be sufficiently randomized at each transfer* such that a user who has seen a token previously cannot identify it when it is transferred to him again (unlinkability). The inadequacy of the schemes discussed above along with two other possibilities, DACs and E-Cash (discussed in section 4.3) is summarized in table 2. Going forward we show how combining DACs and E-Cash gives us a mechanism to satisfy all the properties required by our token scheme.

4.3 Building Blocks

4.3.1 Delegatable Anonymous Credentials

Delegatable anonymous credentials (DACs) is a cryptographic mechanism to delegate access rights repeatedly without revealing the identity of the participants. DACs provide similar functionality as a certificate chain but do not reveal the identity of the intermediate entities of the chain. Towards achieving this, Belinkiy et. al.[36] propose an authentication scheme that creates a tag that authenticates a vector of messages under a secret key. For example, user A_1 can authenticate a set of messages, $\vec{m}$ under his secret key sk^{A_1}. If $\vec{m}$ includes the secret key of another user A_2, sk^{A_2}, the tag becomes a user certificate from A_1 to A_2. The scheme is summarized by the set

Table 3: DAC Cheat Sheet

Algorithm Name	**Description**
$\mathsf{AuthSetup}(1^k)$	Generates groups G_1, G_2, G_T of prime order p whose bit length is proportional to k, a bilinear map $e : G_1 \times G_2 \rightarrow G_T$, and group elements $g, u, u^*, u_1, \cdots, u_n \in G_1$ and $h \in G_2$. It outputs the complete parameter list $par_A = (G_1, G_2, e, p, g, u, u^*, u_1, \cdots, u_n, h)$.
$\mathsf{AuthKg}(par_A)$	Generates $sk \xleftarrow{\$} \mathbb{Z}_p$ and $pk \leftarrow h^{sk}$, and returns (sk, pk).
$\mathsf{Auth}(sk, \vec{m} = (m_1, \cdots, m_n), par_A)$	Generates $K^*, K_1, \cdots, K_n \xleftarrow{\$} \mathbb{Z}_p$. It outputs an authenticator $auth^{sk \rightarrow \vec{m}} = (g^{\frac{1}{sk+K^*}}, h^{K^*}, u^{*K^*}, \{g^{\frac{1}{K^*+K_i}}, h^{K_i}, u_i^{K_i}, g^{\frac{1}{K_i+m_i}}\}_{1 \leq i \leq n})$. The authenticator is used to prove that $\vec{m}$ is authenticated under secret key sk. The need for intermediate keys $K^*, K_1, \cdots, K_n$ and the properties of this authentication mechanism can be found in [36].
$\mathsf{VerifyAuth}(pk, \vec{m} = (m_1, \cdots, m_n), auth^{sk \rightarrow \vec{m}}, par_A)$	Parses $auth^{sk \rightarrow \vec{m}} = (A^*, B^*, C^*, \{A_i, B_i, C_i, D_i\}_{1 \leq i \leq n}$ and verifies $\{e(A^*, pk \cdot B^*) \cdot e(g, h^{-1}) = 1 \wedge e(u^*, B^*) \cdot e(C^*, h^{-1}) = 1 \wedge_{1 \leq i \leq n} (e(D_i, B_i h^{m_i}) \cdot e(g, h^{-1}) = 1)\}$. Returns 1, if all equations match, else 0.

of algorithms shown in table 3. Using Auth, a user A_1 can authenticate the secret key of user A_2. However, since A_2's secret key should not be revealed to A_1, they carry out a secure two party computation (2PC) of the authentication scheme between A_1 and A_2, shown below:

$\mathsf{2PCAuth}(\mathcal{I}(sk_{\mathcal{I}}, \{m_i\}_{1 \leq i \leq l}), \mathcal{O}(pk_{\mathcal{I}}, \{m_i\}_{1 \leq i \leq n}))$ is a secure two party computation between an authentication issuer $\mathcal{I}$ and a message owner $\mathcal{O}$ such that $\mathcal{I}$ does not get any information about $(m_i)_{l+1 \leq i \leq n}$ as well as $\{g^{\frac{1}{K_i+m_i}}\}_{l+1 \leq i \leq n}$.

At the end of $\mathsf{2PCAuth}$, A_2 possesses an authenticator $auth^{A_1 \rightarrow A_2}$ from A_1, which is essentially a certificate on his secret key. Such an authenticator itself is unchanging and therefore reveals the identity of a user. The DAC system uses the notion of user pseudonyms to get around this. In pseudonym systems[105], a user has a single secret key but multiple public keys. User A_2 who has a secret key sk^{A_2}, can choose a random value o and use the commitment $Commit(sk^{A_2}, o)$ as a public key. Different values of o result in different public keys or pseudonyms for the same user. A_2 can be known to user A_1 with public key pk^{A_2} and to user A_3 with public key pk'^{A_2}. Though an adversary cannot link pk^{A_2} and pk'^{A_2}, user A_2 can prove that they are actually commitments to the same secret. In this case, A_1 rather than provide the authenticator

directly, provides a non interactive zero knowledge (NIZK) proof for the authenticator, $\pi^{A_1 \to A_2}$ authenticating the contents of pseudonym pk^{A_2}. The NIZK proof system used is the Groth Sahai proof system[78] which allows the proofs to be (re)randomized everytime they are presented. When A_2 wants to delegate his access right to A_3, he randomizes the pseudonym, proof pair $(pk^{A_2}, \pi^{A_1 \to A_2})$ to $(pk'^{A_2}, \pi'^{A_1 \to A_2})$ where $\pi'^{A_1 \to A_2}$ authenticates the contents of pk'^{A_2}. The new pseudonym, proof pair can be further randomized by users who don't know the underlying contents. This allows user A_3 to once again randomize the credential while delegating to another user A_4 and this ensures that the credential changes each time it is transferred, thus providing strong unlinkability guarantees. The same procedure is followed between each pair of users. Specifically, A_2 will also use its secret key sk^{A_2} to provide an NIZK proof of an authenticator for A_3's secret key sk^{A_3}, $\pi^{A_1 \to A_2}$, which again can be randomized. A_3's credential will then be ($\pi'^{A_1 \to A_2}$, $\pi^{A_2 \to A_3}$) and will be completely unlinkable to the credential that A_2 had, ($\pi^{A_1 \to A_2}$).

Though DACs provide the desirable level of anonymity, the credentials are not single use. When access rights are delegated to a user, the user can repeatedly delegate the rights to any number of other users. If we used just the DAC system in the VoIP setting, a user who has a token can transfer it to a large number of other users, all of whom can call the token issuer, which is undesirable. In addition, once a user has a token, he can reuse it multiple times, regardless of the issuer's interaction experiences with that user. To address this we need to extend DACs to create single use tokens.

4.3.2 E-Cash

Electronic cash refers to mechanisms that allow coins to be exchanged electronically. Typically e-cash schemes contain three entities, the bank, the user and the merchant. The user withdraws coins from the bank and spends them at a merchant who then

deposits the coins at the bank. Like all digital data, coins can be copied and hence e-cash schemes provide a mechanism to prevent double spending. Camenisch et. al.[47] introduced the first efficient anonymous e-cash scheme that identified double spenders without needing the bank to be online for each transaction. To illustrate their e-cash scheme, suppose a bank has a key pair (sk^B, $pk^B = g^(sk^B)$), where g is a generator of some group G of prime order p. Similarly, the user has key pair (sk^U, $pk^U = g^(sk^U)$). A user on coin withdrawal from a bank receives a signature on a set of values (sk^U, s, t) where $s, t \xleftarrow{\$} Z_p$. s is the seed for the serial number and t is the seed for the double spending equation. The serial number is of the form $S \leftarrow g^{\frac{1}{s+x}}$, where $x \in Z_p$. The double spending equation for a coin is of the form $T \leftarrow pk^U \cdot g^{\frac{r}{s+x}}$, where $r \xleftarrow{\$} Z_p$ is chosen by the merchant. If a user U double spends the coin and the merchant(s) chooses two random values r_1 and r_2 for each of the transactions then the two double spending equations $T_1 \leftarrow pk^U \cdot g^{\frac{r_1}{s+x}}$ and $T_2 \leftarrow pk^U \cdot g^{\frac{r_2}{s+x}}$ reveal the identity of the double spender by computing $(\frac{(T_1)^{r_2}}{(T_2)^{r_1}})^{\frac{1}{r_2 - r_1}} = pk^U$.

As seen in table 2, DACs satisfy all our requirements except for allowing single use tokens. The most recent transferable E-Cash scheme[47] uses meta proof techniques which are inefficient. In our scheme, we use serial number and tags from E-Cash to extend DACs and create single use tokens that satisfy all our goals.

4.4 Single Use Anonymous Transferable Token Scheme

A token contains three integral components: (i) the identity of the token issuer, (ii) transfer/call path information, (iii) information that ensures single use and double spender identification. DACs allow us to create tokens where (i) and (ii) satisfy the necessary properties: unforgeablity, verifiablity, privacy and efficiency. To satisfy (iii), in addition to creating single use tokens, we need to ensure the token continues to be unlinkable and therefore privacy preserving. To do this we use techniques from E-Cash to create serial number and tags to make single use tokens. We then show

how these can be made privacy preserving.

4.4.1 Cryptographic Preliminaries

We use the DAC authentication scheme, E-Cash and ElGamal encryption to meet the different requirements of the token framework. These constructs require common parameters, described in ParamGen, that are shared across all users. The clients also need to generate a set of keys for different parts of the scheme, and this is described in KeyGen.

• $\mathsf{ParamGen}(1^k)$ is probabilistic algorithm that outputs the common parameters for the token scheme, par^{TS}. It runs $\mathsf{AuthSetup}(1^k)$(section 4.3.1) to get par^A. Then it generates $\bar{g} \in G_1$ and $\bar{h} \in G_2$ for the ElGamal encryption. It returns $par^{TS} = (par^A, \bar{g}, \bar{h})$. These common parameters are shared by all users of the system and are used for all the token operations.

• $\mathsf{KeyGen}(par^{TS})$ is a probabilistic algorithm that outputs the key pair for a user, A_i, (sk^{A_i}, pk^{A_i}) and is run by each user after they obtain the common parameters. This algorithm parses par^{TS} and uses par^A to generate $(sk'^{A_i}, pk_1'^{A_i}) \leftarrow \mathsf{AuthKg}(par^A)$ (section 4.3.1). Remember $pk_1'^{A_i} \leftarrow h^{(sk_1'^{A_i})}$. It then computes another public key, $pk_2'^{A_i} \leftarrow u^{sk'^{A_i}} (\in G_1)$. The token issue protocol uses these keys. The algorithm then generates $\bar{sk}^{A_i} \xleftarrow{\$} G_1$. It uses this secret key to compute $\bar{pk}_1^{A_i} \leftarrow \bar{g}^{\bar{sk}^{A_i}}$ and $\bar{pk}_2^{A_i} \leftarrow \bar{h}^{\bar{sk}^{A_i}}$. These keys are used for ElGamal encryption whenever A_i is a token issuer. Finally, $sk^{A_i} \leftarrow (sk'^{A_i}, \bar{sk}^{A_i})$ and $pk^{A_i} \leftarrow (pk'^{A_i}, \bar{pk}^{A_i}) \leftarrow ((pk_1'^{A_i}, pk_2'^{A_i}), (\bar{pk}_1^{A_i}, \bar{pk}_2^{A_i}))$.

4.4.2 Construction

In our setting, the token issuer acts like an E-Cash bank. Since, each user is a bank, the situation is analogous to a user behaving like a country with its own currency. As described in E-Cash, the first step in generating the serial number and tags is generating the seeds for them. In the basic DAC scheme, the message vector that an issuer A_1 authenticates contains the secret key of the token recipient, A_2. To this

message vector we now add seeds for the serial number and the tags as shown in figure 10. Specifically, A_1 authenticates the message vector $\vec{m}^{A_2} = \{sk'^{A_2}, s^{A_2}, r^{A_2}, \hat{t}^{A_2}, \check{t}^{A_2}\}$ with his secret key sk'^{A_2}. s^{A_2} is the seed for the serial number and $r^{A_2}, \hat{t}^{A_2}, \check{t}^{A_2}$ are seeds for the tags. This entire procedure essentially forms token issue, at the end of which user A_2 obtains an NIZK proof for the above authenticator from A_1. A_2 then calculates the serial number as $S \leftarrow g^{\frac{1}{sk'^{A_2}+s^{A_2}}}$. Since the serial number is a function of sk'^{A_2} and s^{A_2}, A_2 can prove that it was formed only with secrets authenticated by the token issuer, A_1.

In E-Cash, tags are used to identify a user who tries to double spend a coin(same serial number). These tags need to be created everytime a coin is transferred. A user initiating a token transfer creates partial tags and the user receiving the token completes the tags such that if the same token was transferred twice, the two completed tags are different enough to yield the token duplicator's identity. Unlike E-Cash, we need two sets of tags to determine a double spender for the cases when: a) the token is double spent to two different users, and b) the token is double spent to the same user. Consider the first time an issued token is transferred. In Figure 10, when A_2 transfers the token issued by A_1 to A_3, he creates partial tags of the form, $\hat{T}_1^{A_2} \leftarrow g^{\frac{1}{sk'^{A_2}+\hat{t}^{A_2}}}$, $\hat{T}_2^{A_2} \leftarrow pk_2'^{A_2} \cdot (\hat{T}_1^{A_2})^{r^{A_2}}$ and $\check{T}^{A_2} \leftarrow g^{\frac{1}{sk'^{A_2}+\check{t}^{A_2}}}$. In order to make sure that A_2 does not double spend, A_3 completes the tags by generating a random number, r^{A_3} and then calculating $\hat{T}_{11} \leftarrow \hat{T}_2^{A_2} \cdot (\hat{T}_1^{A_2})^{sk'^{A_3}}$, $\hat{T}_{12} \leftarrow pk_1'^{A_3} \cdot C^{r^{A_2}}$, $\check{T}_{11} \leftarrow pk_2'^{A_2} \cdot (\check{T}^{A_2})^{r^{A_3}}$, and $\check{T}_{12} \leftarrow h^{r^{A_3}}$. The public key that satisfies the following equation is the public key of the double spender.

$$\frac{e(\hat{T}_{l1}, \hat{T}'_{l2})}{e(\hat{T}'_{l1}, \hat{T}_{l2})} = e(pk^{DS}, \frac{\hat{T}'_{l2}}{\hat{T}_{l2}}) \tag{2}$$

To see why this is the case, consider a user A_i who transfers the same token to

both A_j and A_k.

For A_j ,

$$\hat{T}_{l1} \leftarrow pk_2'^{A_i} \cdot (\hat{T}_1^{A_i})^{sk'^{A_j}+r^{A_i}} \qquad \hat{T}_{l2} \leftarrow (h)^{sk'^{A_j}+r^{A_i}}$$

For A_k ,

$$\hat{T'}_{l1} \leftarrow pk_2'^{A_i} \cdot (\hat{T}_1^{A_i})^{sk'^{A_k}+r^{A_i}} \qquad \hat{T'}_{l2} \leftarrow (h)^{sk'^{A_k}+r^{A_i}}$$

Note that the random values, $r^{A_2}, \hat{t}^{A_2}, \check{t}^{A_2}$ used to generate the token tags will be the same when A_i transfers the token to both A_j and A_k. Plugging in these values into the left hand side of equation 2 gives:

$$\begin{aligned}
\frac{e(\hat{T}_{l1}, \hat{T'}_{l2})}{e(\hat{T'}_{l1}, \hat{T}_{l2})} &= \frac{e(pk_2'^{A_i} \cdot (\hat{T}_1^{A_i})^{sk'^{A_j}+r^{A_i}}, (h)^{sk'^{A_k}+r^{A_i}})}{e(pk_2'^{A_i} \cdot (\hat{T}_1^{A_i})^{sk'^{A_k}+r^{A_i}}, (h)^{sk'^{A_j}+r^{A_i}})} \\
&= \frac{e(pk_2'^{A_i}, (h)^{sk'^{A_k}+r^{A_i}})}{e(pk_2'^{A_i}, (h)^{sk'^{A_j}+r^{A_i}})} \\
&= e(pk_2'^{A_i}, \frac{\hat{T'}_{l2}}{\hat{T}_{l2}})
\end{aligned}$$

From this, $pk^{DS} = pk_2'^{A_i}$ and thus, A_i will be correctly identified as the double spender. This explains the need for the first set of tags, $\hat{T}_{l1}$ and $\hat{T}_{l2}$. The second set of tags helps catch the token double spender if he transfers the same token twice to the same user. This is useful, as users never need to store the details of a token once they have transferred it. In this case, consider the user A_i who transfers the same token twice to A_j. The first set of tags will both be of the form $(pk_2'^{A_i} \cdot (\hat{T}_1^{A_i})^{sk'^{A_j}+r^{A_i}}$, $(h)^{sk'^{A_j}+r^{A_i}})$. On the other hand, the second set of tags, ($\check{T}_{l1}$, $\check{T}_{l2}$) will be different as A_j generates a new random number, r^{A_j} for each token transfer. In this case a similar equation to equation 2 can be used to identify A_i as the double spender, and is shown in equation 3.

$$\frac{e(\check{T}_{l1}, \check{T'}_{l2})}{e(\check{T'}_{l1}, \check{T}_{l2})} = e(pk^{DS}, \frac{\check{T'}_{l2}}{\check{T}_{l2}}) \tag{3}$$

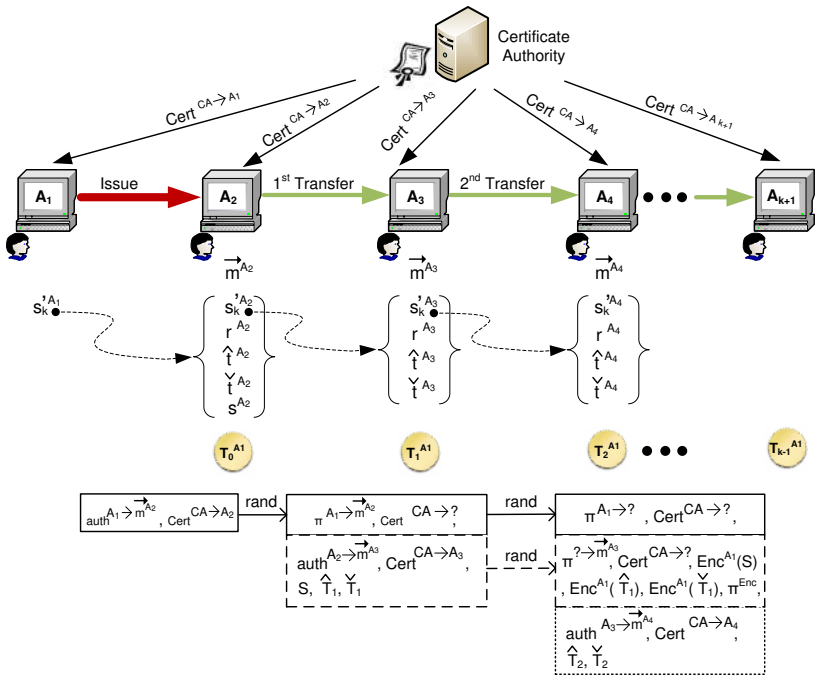


Figure 10: Single Use Anonymous Transferable Token Scheme. Question marks are used to indicate that an identity has been anonymized.

As shown in figure 10, after the first transfer, the token contains, a serial number and the tags representing the first transfer. When this token needs to be transferred again (second transfer), the serial number and tags need to be randomized. In addition, the issuer, A_1 needs to be able to retrieve the original serial number and the tags, to detect and catch a double spender. To satisfy these requirements, we encrypt the serial number and tags with the public key of the issuer, $\bar{pk}^{A_1}$, a technique introduced in [48]. As shown in figure 10, when A_3 transfers the token to A_4, $(S, \hat{T}_{11}, \hat{T}_{12}, \check{T}_{11}, \check{T}_{12})$ gets encrypted to $(Enc^{A_1}(S),\ Enc^{A_1}(\hat{T}_{11}),\ Enc^{A_1}(\hat{T}_{12}),\ Enc^{A_1}(\check{T}_{11}),\ Enc^{A_1}(\check{T}_{12}))$, the two sets being unlinkable with each other. The serial number and tags can be encrypted each time, maintaining the unlinkability of tokens. We use the ElGamal encryption as the token issuer requires a single decryption operation even if the contents (serial number and tags) are encrypted multiple times. Specifically, $Enc^{A_1}(S) \leftarrow ((\bar{pk}_2^{A_1})^{\bar{r}_1} \cdot S, \bar{g}^{\bar{r}_1})$, $Enc^{A_1}(\hat{T}_{11}) \leftarrow ((\bar{pk}_2^{A_1})^{\bar{r}_2} \cdot S, \bar{g}^{\bar{r}_2})$,

$Enc^{A_1}(\hat{T}_{12}) \leftarrow ((\bar{pk}_1^{A_1})^{\bar{r}_3} \cdot S, \bar{g}^{\bar{r}_3})$, $Enc^{A_1}(\check{T}_{11}) \leftarrow ((\bar{pk}_2^{A_1})^{\bar{r}_4} \cdot S, \bar{g}^{\bar{r}_4})$, $Enc^{A_1}(\check{T}_{12}) \leftarrow ((\bar{pk}_1^{A_1})^{\bar{r}_5} \cdot S, \bar{g}^{\bar{r}_5})$, where $\bar{r}_1, \cdots, \bar{r}_5 \xleftarrow{\$} \mathbb{Z}_p$.

Encrypting the serial number and tags makes it hard to check if they have been generated correctly. To resolve this, we attach the ZK proofs that show that the encrypted serial number and tags are generated correctly. For example, for $Enc^{A_1}(S)$ whose first term is of the form $P \cdot S$, A_3 generates commitments of P and S, C^P and C^S respectively. It creates a proof to shows C^P and C^S have been generated correctly and a proof that the multiplication of C^P and C^S is the commitment to $P \cdot S$. Similar proofs are generated for the other encrypted values. Using the Groth Sahai proof system[78] we concatenate all of the proofs generated, into one final proof, π_1^{Enc}. We note that the authenticator information also shown in figure 10 is randomized using DACs as discussed in section 4.3.1

Finally for transfers beyond the second transfer, the first set of serial number and tags needs to be randomized again to preserve unlinkability. The ciphertexts encrypting the serial number and tags after the second transfer are of the form $(A = (\bar{g}^x)^r \cdot m, B = \bar{g}^r)$, where $\bar{g}^x$ is the public key of the decryptor. Then a user who has the ciphertext and the decryptor's public key can re-encrypt it again by computing $((\bar{g}^x)^{r'} \cdot A, \bar{g}^{r'} \cdot B)$. We can then modify the commitments and the proofs according to the new random value r' using the Groth Sahai proof system[78].

The above discussion shows how the serial number for the token and double spender tags to catch user A_2 can be created and randomized at each transfer. However we need to create tags for each new user in the transfer path. When A_3 decides to transfer the token we need to create similar tags for him too. In this case, A_2 creates an authenticator for A_3's secret key and the the tag seeds as shown in figure 10. The randomization procedure for these new tags is exactly similar. This concludes the token transfer operation. Submitting a token involves randomizing all the components of the token and submitting it to the issuer. There is no need to carry

out the DAC authentication scheme in this case.

Our construction is almost complete except for one final subtlety. In the DAC setting, users have one secret key and many public keys. In order to catch double spenders we need to have at least one of these public keys registered with a certificate authority (CA). In VoIP systems, the authentication server of VoIP providers like Skype, Google Talk and Vonage can play the role of the CA. In fact, in Skype, user accounts are already associated with a public key, which they use for communication. We, however, need a new certificate issuing protocol because a conventional certificate reveals the identity of the certificate holder and we need one that reveals the identity only in direct interaction with a user and when the user behaves dishonestly (for example, double spends a token). To do this we can once again use the DAC authentication scheme with a CA providing an NIZK proof of an authenticator for each user as shown in figure 10. A user on transferring a token also adds his certificate. Since this certificate can also be randomized it continues to maintain unlinkability of the token. On randomization the certificate no longer reveals the identity of the user but only shows that the certificate has been generated by the CA.

With this knowledge, the components of a token are (see figure 10): (1) the identity of the token issuer, (2) randomized certificates for all users so far in the transfer path, (3) a chain of DAC NIZK proof of authenticators which validates the actual transfer/call path information, and (4) randomized serial number information + randomized tag information for all previous transfers + tag information for the current transfer. Other than the identity of the token issuer, the remaining content of the token is randomizable everytime it is transferred. To summarize, our construction provides a way to (a) use the DAC authentication scheme to create certificates and generate seeds for the serial number and double spender tags, (b) create serial number and tags from these seeds, (c) identify a double spender, and (d) randomize the serial number and tag information at each transfer.

4.4.3 Scheme Definition

We formalize the algorithms that define our token scheme as follows:

- $\mathsf{ParamGen}(1^k)$ is probabilistic algorithm that outputs the system parameters par^{TS}.

- $\mathsf{KeyGen}(par^{TS})$ is a probabilistic algorithm that outputs the key pair of user, A_i: (sk^{A_i},pk^{A_i}). This pair represents all the keys that are generated.

- $\mathsf{IssueToken}(A_1(sk^{A_1}, pk^{A_2}), A_2(sk^{A_2}, pk^{A_1}))$ is an interactive protocol where A_1 issues a token to A_2. After this protocol ends, A_1 gets either its view $\mathcal{V}_{A_1}^{\mathsf{issue}}$ or $\perp$, and A_2 gets either a token $\mathsf{Token}_0^{A_1}$ or $\perp$.

- $\mathsf{TransferToken}(A_i(sk^{A_i}, pk^{A_1}, pk^{A_{i+1}}, \mathsf{Token}_{i-2}^{A_1}),$
  $A_{i+1}(sk^{A_{i+1}}, pk^{A_1}, pk^{A_i}))$ is an interactive protocol between A_i and A_{i+1}. pk^{A_1} is the public key of the issuer of $\mathsf{Token}_{i-2}^{A_1}$'. At the end, A_i has its view $\mathcal{V}_{A_i}^{\mathsf{transfer}}$ or $\perp$, and A_{i+1} has either a token $\mathsf{Token}_{i-1}^{A_1}$ or $\perp$.

- $\mathsf{SubmitToken}(A_{k+1}(sk^{A_{k+1}}, pk^{A_1}, \mathsf{Token}_{k-1}^{A_1}),$
  $A_1(sk^{A_1}, pk^{A_{k+1}}, D^{A_1}))$ is an interactive protocol between A_{k+1} and A_1. A_1 will accept $\mathsf{Token}_{k-1}^{A_1}$ if it was correctly issued by A_1 and has never been submitted before. D^{A_1} represents A_1's token database. At the end of this protocol, A_{k+1} gets either its view $\mathcal{V}_{A_{k+1}}^{submit}$ or $\perp$, and A_1 gets either an updated list D'^{A_1}, or two tokens $\mathsf{Token}_{k+1}^{A_1}$ and $\mathsf{Token}_{l}^{A_1}$ which have the same serial number, or $\perp$.

- $\mathsf{Identify}(\mathsf{Token}_{l}^{A_1}, \mathsf{Token}_{l'}^{A_1})$ is a deterministic algorithm. If both $\mathsf{Token}_{l}^{A_1}$ and $\mathsf{Token}_{l'}^{A_1}$ come from the same $\mathsf{Token}_0^{A_1}$, it outputs the public key of the token double spender. Otherwise it returns $\perp$.

- $\mathsf{VerifyGuilt}(pk^{A_i}, \Pi)$ is a deterministic algorithm which outputs 0 if Π is a correct proof that the owner of pk^{A_i} double spent the token, or 1 otherwise.

Theorem 4.4.1 *Protocols* ParamGen, KeyGen, IssueToken, TransferToken, SubmitToken, Identify, *and* VerifyGuilt *achieve correctness, unforgeability, double spender identification and anonymity assuming* HSDH, $\mathsf{BB}-\mathsf{HSDH}$, $\mathsf{BB}-\mathsf{CDH}$, *and* SXDH

Most of the proofs follow from the underlying schemes, namely DAC and e-cash, except anonymity. We need to define a new anonymity game analogous to the one in [48]. In Canard et. al.[48], the adversary, *Adv* runs the e-cash credential transfer protocol (spending protocol) with a challenged user i_b, where b could be either 0 or 1, and has to determine b. In our case, since the identity of a user is known in a direct interaction, *Adv* can easily win the same game. We, therefore, modify the game such that the challenged user i_b runs the token transfer protocol with an intermediate user A_j first. A_j, then, transfers it to *Adv*, who then tries to determine b. This game captures the concept of *interaction anonymity* where the concern is the privacy of previous interactions.

4.5 Security Evaluation

4.5.1 Algorithms and protocols

We formalize and summarize the algorithms described in the previous sections as follows:

- $\mathsf{ParamGen}(1^k)$ is probabilistic algorithm that outputs the system parameters par^{TS}.

- $\mathsf{KeyGen}(par^{TS})$ is a probabilistic algorithm that outputs the key pair of user, A_i: (sk^{A_i},pk^{A_i}). This pair represents all the keys that are generated.

- $\mathsf{IssueToken}(A_1(sk^{A_1}, pk^{A_2}), A_2(sk^{A_2}, pk^{A_1}))$ is an interactive protocol where A_1 issues a token to A_2. After this protocol ends, A_1 gets either its view $\mathcal{V}_{A_1}^{\mathsf{issue}}$ or $\perp$, and A_2 gets either a token $\mathsf{Token}_0^{A_1}$ or $\perp$.

- $\mathsf{TransferToken}(A_i(sk^{A_i}, pk^{A_1}, pk^{A_{i+1}}, \mathsf{Token}_{i-2}^{A_1}), A_{i+1}(sk^{A_{i+1}}, pk^{A_1}, pk^{A_i}))$ is an interactive protocol between A_i and A_{i+1}. pk^{A_1} is the public key of the issuer of $\mathsf{Token}_{i-2}^{A_1}$'. At the end, A_i has its view $\mathcal{V}_{A_i}^{\mathsf{transfer}}$ or $\perp$, and A_{i+1} has either a token $\mathsf{Token}_{i-1}^{A_1}$ or $\perp$.

- $\mathsf{SubmitToken}(A_{k+1}(sk^{A_{k+1}}, pk^{A_1}, \mathsf{Token}_{k-1}^{A_1}), A_1(sk^{A_1}, pk^{A_{k+1}}, D^{A_1}))$ is an interactive protocol between A_{k+1} and A_1. A_1 will accept $\mathsf{Token}_{k-1}^{A_1}$ if it was correctly issued by A_1 and has never been submitted before. D^{A_1} represents A_1's token database. At the end of this protocol, A_{k+1} gets either its view $\mathcal{V}_{A_{k+1}}^{submit}$ or $\perp$, and A_1 gets either an updated list D'^{A_1}, or two tokens $\mathsf{Token}_{k+1}^{A_1}$ and $\mathsf{Token}_{l}^{A_1}$ which have the same serial number, or $\perp$.

- $\mathsf{Identify}(\mathsf{Token}_{l}^{A_1}, \mathsf{Token}_{l'}^{A_1})$ is a deterministic algorithm. If both $\mathsf{Token}_{l}^{A_1}$ and $\mathsf{Token}_{l'}^{A_1}$ come from the same $\mathsf{Token}_{0}^{A_1}$, it outputs the public key of the token double spender. Otherwise it returns $\perp$.

- $\mathsf{VerifyGuilt}(pk^{A_i}, \Pi)$ is a deterministic algorithm which outputs 0 if Π is a correct proof that the owner of pk^{A_i} double spent the token, or 1 otherwise.

4.5.2 Correctness

We say a token submit is correct if an honest issuer gets an updated database as part of running protocol SubmitToken with the token submitter, only when the submitter submits a valid token. We say that a token issue and token transfer are correct if a honest user gets a valid token by running IssueToken or TransferToken protocol respectively, such that the token can be submitted or transferred and the submitter on running SubmitToken with the issuer, will never have the issuer outputting $\perp$.

4.5.3 Security and anonymity

This section shows the security and anonymity model that any token transfer scheme needs to satisfy. It then provides the security proofs of our token transfer scheme

under this model.

4.5.3.1 Definition of oracles

We follow a similar approach as [48]. Suppose that the parameter par^{TS} is given to the oracles. All the users' public keys and secret keys are initially created and managed by the oracles in databases PK and SK. They also manage the set of views of tokens. There are three tables IT, OT and ST. The tokens issued from the oracles are stored in IT, those issued to, or transferred from or to the oracles in OT, and those submitted to the oracles in ST. To evaluate the security of our scheme we use the following oracles:

- OCreateUser(i) executes KeyGen(par^{TS}) and stores the output public key pk^{A_i} in PK[i] and the secret key sk^{A_i} in SK[i].
- OCorrupt(i) outputs sk^{A_i} and sets SK[i] $= \perp$. When an adversary executes this oracle he gets all of A_i's tokens. After this protocol is run, the adversary can act as A_i as well as any of the other users that he has corrupted.
- OIssueI(pk^{A_1}, pk^{A_2}) runs IssueToken protocol playing the token issuer. The adversary should have the secret key sk^{A_2} to execute this oracle. The oracle stores $\mathcal{V}_{A_1}^{\mathsf{issue}}$ in IT[1].
- OIssueU(pk^{A_1}, pk^{A_2}) runs IssueToken playing the token receiver, A_2's side. The adversary should have sk^{A_1} to execute this oracle. The oracle stores the resulting token in OT[2]
- OIssueI&U(pk^{A_1}, pk^{A_2}) runs IssueToken protocol playing both the token issuer and receiver. If the result of the protocol is $\mathcal{V}_{A_1}^{\mathsf{issue}}$ and $\mathsf{Token}_0^{A_1}$, they are stored in IT[1] and OT[2], respectively. The adversary should have neither sk^{A_1} nor sk^{A_2}.

- $\mathsf{OTransferS}(pk^{A_i}, \mathsf{Token}_{i-2}^{A_1}, pk^{A_{i+1}})$ runs $\mathsf{TransferToken}$ protocol playing the user who is transferring the token. The adversary should have secret key $sk^{A_{i+1}}$ to execute this oracle. If $\mathsf{OT}[i]$ does not have the token, the protocol is aborted. If the protocol is successful then $\mathsf{Token}_{i-2}^{A_1}$ is removed from $\mathsf{OT}[i]$ and sent to the adversary. $\mathsf{OT}[i]$ is updated with the view $\mathcal{V}_{A_i}^{transfer}$.

- $\mathsf{OTransferR}(pk^{A_i}, \mathsf{Token}_{i-2}^{A_1}, pk^{A_{i+1}})$ runs $\mathsf{TransferToken}$ protocol playing the token receiver, A_{i+1}'s side. The adversary should have sk^{A_i} and $\mathsf{Token}_{i-2}^{A_1}$ before executing this oracle. If the protocol completes successfully, the resulting $\mathsf{Token}_{i-1}^{A_1}$ is stored in $\mathsf{OT}[i+1]$.

- $\mathsf{OTransferS\&R}(pk^{A_i}, \mathsf{Token}_{i-2}^{A_1}, pk^{A_{i+1}})$ runs $\mathsf{TransferToken}$ protocol playing both sides. If $\mathsf{OT}[i]$ does not have the token, the protocol is aborted. Otherwise, after running the protocol, $\mathsf{Token}_{i-2}^{A_1}$ is removed from $\mathsf{OT}[i]$ and sent to A_{i+1}. $\mathsf{Token}_{i-1}^{A_1}$ is now stored in $\mathsf{OT}[i+1]$. A_i's output, $\mathcal{V}_{A_i}^{transfer}$ is now stored in $\mathsf{OT}[i]$.

- $\mathsf{OSubmitS}(pk^{A_{k+1}}, \mathsf{Token}_{k-1}^{A_1}, pk^{A_1})$ runs $\mathsf{SubmitToken}$ protocol playing A_{k+1}. The adversary should have sk^{A_1} to execute this oracle. If the protocol is not aborted $\mathsf{OT}[k+1]$ is updated with A_{k+1}'s view of the protocol, $\mathcal{V}_{A_{k+1}}^{submit}$. If $\mathsf{SubmitToken}$ outputs $\mathsf{Token}_l^{A_1}, \mathsf{Token}_{k-1}^{A_1}$, it runs $\mathsf{Identify}(\mathsf{Token}_l^{A_1}, \mathsf{Token}_{k-1}^{A_1},)$ and outputs the resulting public key.

- $\mathsf{OSubmitR}(pk^{A_{k+1}}, \mathsf{Token}_{k-1}^{A_1}, pk^{A_1})$ runs $\mathsf{SubmitToken}$ protocol playing the issuer's side. The adversary should have both $\mathsf{Token}_{k-1}^{A_1}$ and $sk^{A_{k+1}}$ to run this oracle. sk^{A_1} should not belong to the adversary. It updates ST if the protocol completes successfully. If $\mathsf{SubmitToken}$ outputs $\mathsf{Token}_l^{A_1}, \mathsf{Token}_{k-1}^{A_1}$, it runs $\mathsf{Identify}(\mathsf{Token}_l^{A_1}, \mathsf{Token}_{k-1}^{A_1},)$ and outputs the resulting public key.

4.5.3.2 Unforgeability

As in [48], the unforgeability requirement reduces to the fact that any set of users should not be able to spend more tokens than those issued or transferred to them.

Game. Suppose an adversary *Adv* is a probabilistic polynomial-time Turing Machine that has access to all of the user's public keys in PK and $par^{TS} \leftarrow \mathsf{ParamGen}(1^k)$. *Adv* can play with the oracles OCreateUser, OCurrupt, OIssuel, OIssuel&U, OTransferS, OTransferR, OTransferS&R and OSubmitR, as many times as he wants. *Adv* wins the game if $q_I + q_R < q_S$ where q_I is the number of successful queries to the oracle OIssuel, q_R is the number of successful queries to the oracle OTransferS, and q_S is the number of successful queries to the oracle OTransferR.

Theorem 4.5.1 *The proposed scheme is unforgeable.*

Proof: Suppose the adversary, *Adv* succeeds in forging a token in the unforgeability game. This means *Adv* produces at least one new token that is acceptable by the oracle OTransferR. Based on the number of transfers that the token has undergone, we can divide this into three cases. If the new token is a directly issued token, then the entire token is essentially a delegatable anonymous credential. The existence of the new token means breaking F-unforgeability[36], which is a contradiction based on the computational assumption in [36]. If the new token has undergone a single transfer then it consists of the delgatable anonymous credential, a serial number, and a tag. The existence of the new token then breaks F-unforgeability, or violates the weak BB assumption [39]. Based on the assumptions in [36], this is infeasible. The final case is where the new token has undergone more than one transfer. In this case, the new token is a GS-NIZK proof. Because of the extractability of the GS-NIZK proof, we can extract the witness of the proof. Thus, like the second case, we can show that this means breaking the F-unforgeability or violating the weak BB assumption. Therefore, the proposed scheme is unforgeable. ∎

4.5.3.3 Anonymity

For the token scheme to be privacy preserving, in our setting, we need it to have strong anonymity guarantees. In this section we define the exact anonymity requirements, and call it *interaction anonymity*. We define the *interaction anonymity* game analogous to the one in [48]. In [48], the adversary, Adv runs the e-cash credential transfer protocol (spending protocol) with a challenged user i_b, where b could be either 0 or 1, and has to determine b. In our case, since the identity of a user is known in a direct interaction, Adv can easily win the same game. We, therefore, modify the game such that the challenged user i_b runs the token transfer protocol with an intermediate user A_j first. A_j, then, transfers it to Adv, who tries to determine b. This game captures the concept of *interaction anonymity* where the concern is the privacy of previous interactions. We have previously used A_* to define all our users. We use i_0 and i_1 to maintain a similar notation as [48], enabling us to highlight the difference between the two anonymity games. i_0 and i_1 could represent any two users. We define the anonymity game more precisely as follows:

$\mathbf{Game}^{\mathsf{anonymity}}(1^k)$

1 $par^{TS} \xleftarrow{\$} \mathsf{ParamGen}(1^k)$

2 SK, PK, IT, OT, and ST are created.

3 $pk^{i_0}, pk^{i_1}, pk^{A_j}, \mathsf{Token}^{A_1}, \mathsf{Token}'^{A_1} \xleftarrow{\$} Adv^{\mathsf{SetofOracles}}$,

where $\mathsf{SK}[1] \neq \perp$, $\mathsf{SK}[j] \neq \perp$, $\mathsf{SK}[i_0] \neq \perp$, $\mathsf{SK}[i_1] \neq \perp$.

$\mathsf{Token}^{A_1}, \mathsf{Token}'^{A_1}$ have the same length.

4 Suppose Token^{A_1} belongs to A_m, and Token'^{A_1} belongs to A_n,

where both users could be corrupted by Adv.

$\mathsf{OTransferR}(pk^{A_m}, pk^{i_0}, \mathsf{Token}^{A_1})$ and $\mathsf{OTransferR}(pk^{A_n}, pk^{i_1}, \mathsf{Token}'^{A_1})$ are executed.

5 $b \xleftarrow{\$} \{0,1\}$ and $\mathsf{OTransferS\&R}(pk^{i_b}, pk^{A_j})$ is executed.

6 $\mathsf{OTransferS}(pk^{A_j}, pk^{A^{Adv}})$ is executed, where A^{Adv} can be any user who is corrupted by Adv.

7 $b' \leftarrow Adv^{\mathsf{SetofOracles}'}$

8 If $b = b'$ return 1. Else, return 0.

(*) Adv cannot use $\mathsf{OSubmitR}$ more than once for each token

Token^{A_1} and Token'^{A_1} through the whole experiment,

even when they are transferred to other users controlled by oracles.

(*) $\mathsf{SetofOracles}$ means Adv can play with all the oracles.

(*) $\mathsf{SetofOracles}'$ means Adv can play with all oracles except

$\mathsf{OTransferS}(pk^{i_0}, \mathsf{Token}^{A_1}, \cdot)$, $\mathsf{OTransferS}(pk^{i_1}, \mathsf{Token}'^{A_1}\cdot)$,

$\mathsf{OSubmitS}(pk^{i_0}, \mathsf{Token}^{A_1}, A_1)$, and $\mathsf{OSubmitS}(pk^{i_1}, \mathsf{Token}'^{A_1}, A_1)$ are not allowed.

In the above game, the following inequality should hold for a scheme if it has to meet *interaction anonymity*:

$$|Pr[\mathbf{Game}^{\mathsf{anonymity}}(1^k) = 1] - Pr[\mathbf{Game}^{\mathsf{anonymity}}(1^k) = 0]| < \frac{1}{p(k)}$$

Theorem 4.5.2 *The proposed scheme preserves interaction anonymity.*

Proof: From the anonymity experiment, the token has undergone at least 3 transfers after which the adversary, Adv needs to determine whether i_0 or i_1 owned it previously. This means that the token is composed of GS-Proofs, a serial number, and tags. The serial number and tags are encrypted with the issuer's public key. The harder case is if the adversary has seen the token before, by corrupting users A_m and A_n. Since the GS-proofs are randomized [36], and the serial number and tags are re-encrypted with a new random number at every transfer, both of which ensure unlinkability, Adv cannot link the token that he obtains to any of the tokens that he previously owned. More precisely speaking, the randomizability of GS-proofs[78] shows that the randomized GS-proof cannot be distinguishable from a simulated GS-proof that is generated based on simulated parameters even though the adversary knows the trapdoor information of the proof. This means the GS-proofs that were part of the token owned by the Adv are unlinkable to the GS-proof in the token that he obtains at the end of the experiment. As far as the serial number and tags are concerned, any of the re-encrypted Elgamal ciphertexts are indistinguishable from the two random element tuple (g^{r_1}, g^{r_2}), where $r_1, r_2 \xleftarrow{\$} \mathbb{Z}_q$ based on DDH assumption. Therefore, the serial number and tags previously seen by the adversary are unlinkable to the ones that are part of the token that he obtains at the end of the experiment. The proposed scheme therefore preserves *interaction anonymity.* ∎

4.5.3.4 Identification of double spender

No user can double spend or transfer a token twice without revealing his identity. We define this requirement through the following game:

Game. Let an adversary Adv be a polynomial probabilistic Turing Machine that has access to all of the users' public keys in PK and par^{TS}. Adv can play any number of times with all of the oracles. Then Adv chooses a challenge token $Token$ that belongs to one of the users that he has corrupted, A_i. After that, Adv uses $Token$ twice

using either OTransferR or OSubmitR. Adv can again play with all of the oracles any number of times. Adv wins the game if, on running OSubmitR, the Submit protocol outputs $Token', Token''$, where both tokens come from $Token$, and the output of Idenitfy$(Token', Token'')$ is not a public key whose secret key is $\perp$ in SK.

Theorem 4.5.3 *The proposed scheme identifies double spenders.*

Proof: We divide the double spending into two cases. The first case is where a user A_i transfers his token to two different users, A_m and A_n. A_m and A_n use their public keys to make the first set of tags ($\hat{T}_{l1}$, $\hat{T}_{l2}$). Therefore, these two tags are different ensuring that when the issuer receives both these tokens, no matter how many transfers the tokens have undergone, the double spending will be detected as shown in equation 2. The only way the tags are not different is if A_m and A_n use the same public key, which is not possible, as for them to be regarded as different entities, their (registered) public keys should be different from each other. If they are the same entity the situation is considered in the next case.

The second case is if A_i transfers the same token twice, as the receiving user does not have access to the serial number of the token he receives. In this case, the second set of tags ($\check{T}_{l1}$, $\check{T}_{l2}$) for the two copies of the double spent token are different as the receiving user will use a different random number for each interaction. When these tokens are submitted to the issuer, ($\check{T}_{l1}$, $\check{T}_{l2}$) will reveal the double spender, A_i as shown in equation 3. A_i can transfer the token to another user corrupted by Adv, A_j who uses the same random number for both interactions. A_j will then transfer these tokens to some other set of users. However when these tokens are submitted to the issuer, A_j will instead be caught as the double spender. ∎

4.6 Implementation and Evaluation

Our scheme is built on recent cryptographic primitives (DAC - Crypto 2009 and the underlying GS Proof System - Crypto 2008), and we are not aware of implementations that exist for them. We, therefore, first built these primitives, in C, on top of the PBC library[24] which performs pairing based mathematical operations. We use type 'D' MNT curves with group order of 159-bit length. We then implemented the different algorithms of our token scheme on top of these primitives. Since the token scheme implements several cryptographic primitives, we first study it's performance with respect to the time taken and the message lengths (network bandwidth) generated by each of its protocols.

4.6.1 Operation Costs

Startup Costs: From a cryptographic standpoint there are two primary entities in our system, the certificate authority (CA) and the user (client). Each of these entities on startup perform a particular set of operations. The CA on startup generates common parameters and then generates it's keys using these common parameters. A client on startup obtains the common parameters from the CA, generates its keys and finally gets them certified by the CA. The cost of each of these startup operations were measured on an Intel Xeon 5160 with a 3 GHz processor. Each operation was run 10 times and the mean and standard deviation values were calculated and the results are shown in figures 11 and 12. Figure 11 shows the time taken by each operation, and figure 12 measures the lengths of messages that need to be passed over the network (bandwidth utilized). Generating the common parameters includes generating the bilinear groups and all the group generators as described in ParamGen in section 4.4.1. From the figures, the block marked *Common Parameter Generation* shows that it takes around 240 ms to generate these parameters and they can be encapsulated in a message of length $\approx$ 4 KB. All clients (including the CA) on

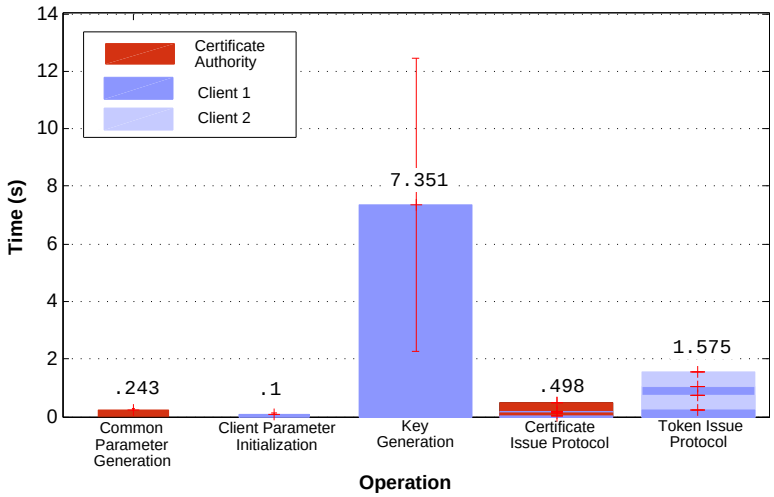


Figure 11: Time - Operation Preliminaries.

startup contact the CA to obtain this 4 KB message and use it to generate their secret keys and their public keys, (sk'^{A_i}, pk'^{A_i}) and $(\bar{sk}^{A_i}, \bar{pk}^{A_i})$ as described in the algorithm KeyGen in section 4.4.1. The key generation is a costly operation as seen in the block marked *Key Generation* and takes on average $\approx 7\,s$ with a standard deviation of $\approx 5\,s$. At this point the CA is done with its startup costs.

The clients still need to certify their public keys pk'^{A_i}, from the CA. Towards this, the client and the CA engage in a 2PC protocol for creating the NIZKPK proof of the authenticator [36]. In figures 11 and 12, *Certificate Issue Protocol* shows the time taken and the lengths of the messages exchanged between the client and the CA are shown in the block marked *Certificate Issue Protocol*. All the messages are under $2\,KB$, and the overall time of the operation notwithstanding network latencies is roughly $.5\,s$. After this, the clients are ready interact with each other either issuing, transferring or submitting tokens. The total startup time for a client on average is $\approx 8\,s$.

Cost of Token Operations: After the one time startup cost, clients can engage in token issue, transfer and submit. The token issue protocol is a three way exchange between the token issuer and the receiver. They carry out a secure 2PC of the issuer

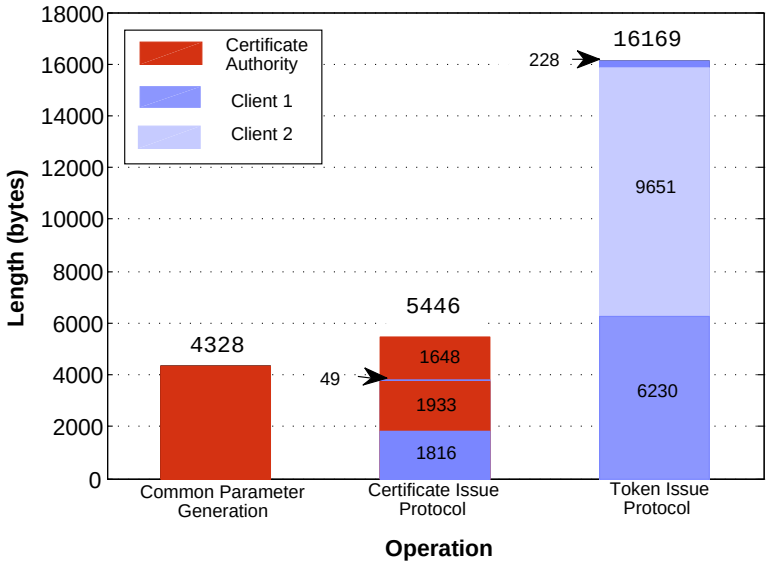


Figure 12: Length - Operation Preliminaries

authenticating the receiver's secrets as described in IssueToken. The time taken and the length of messages generated are depicted by the block *Token Issue* in figures 11 and 12. A token issue takes on average $1.5\,s$ to complete. Once a token is issued the receiving client can submit it back or transfer the token further. The cost of submit or transfer operations varies based on the number of times the token has been transferred. The length of a token being submitted or transferred based on the number of previous transfers is shown in figure 14. Belinkiy et. al. [36] were the first to introduce DACs with proof size that increased linearly in the number of hops from the token issuer. To that we add E-Cash tags (the serial number is constant size) which also grow linearly with the number of transfers. Figure 14 corroborates this showing a linear increase in size of the token based on the number of transfers it has undergone. Furthermore, the close similarity between the length of a token being submitted after L transfers ($L \geq 2$) and a token being transferred $L+1$ hops, is due to the fact that, in order to submit an L hop transferred token, the owner needs to randomize the token in the exact fashion as a token transfer. For tokens being submitted, the token lengths increases by $15\,KB$ each transfer. This is important

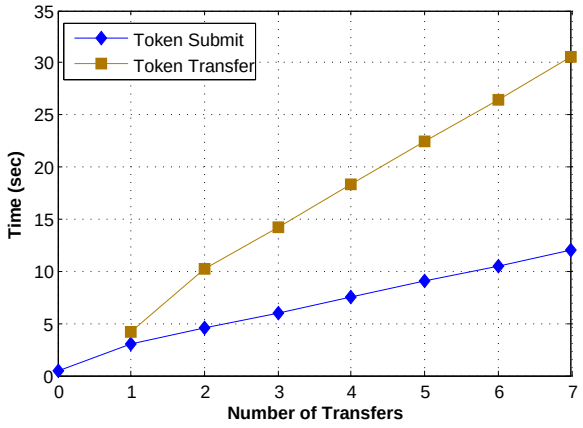


Figure 13: Time - Coin Transfer and Submit

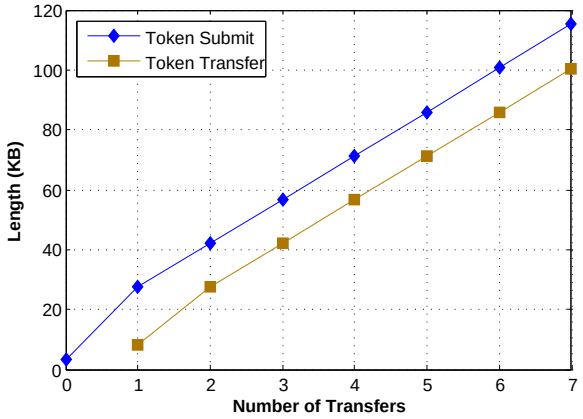


Figure 14: Length - Coin Transfer and Submit

as we plan to use this framework in a VoIP setting where call signaling and call teardown is performed over both UDP and TCP. In IPv4, for UDP, the maximum packet size is 65507 which implies for UDP based applications, without splitting the packets, we restrict ourselves to only allowing 4 hops/3 transfer tokens (size 58 KB). For UDP, a token can therefore be transferred at most 3 times after which it must be submitted. We note that the size of the token being submitted when there are no transfers is significantly lower ($\approx 3\,KB$ for 0 transfer, compared to $\approx 28\,KB$ for 1 transfer) because we optimize away the randomizations and perform them only when the token is first transferred.

Figure 13 depicts the most significant time activity for token transfers and submits,

averaged over 10 runs (low variance). When a token is submitted the receiver only needs to validate the token. On the other hand, when a token is transferred, a 2PC computation needs to be carried out for generating the authenticator and the double spending tags. Due to this the overall time taken for token transfer is significantly more than that of token submit. In figure 13 we see that the time taken for token submit increases linearly by $1.5\,s$ per hop and that of token transfers by $4\,s$ per hop. These values will dictate feasibility of the token framework in a particular application setting. In the next section we discuss the implications of these costs in a VoIP setting and analyze its performance with respect to a spammer threat model.

4.6.2 Applying The Token Scheme To Prevent VoIP Spam

We assume the call setup and teardown signaling is provided by the Session Initiation Protocol (SIP)[127]. For two users to communicate with each other using SIP, they need to know each other's SIP Uniform Resource Identifier (similar to an email id). SIP then uses a three-way handshake mechanism to establish a call and a two way handshake to teardown a call. We piggyback our token mechanisms on top of the SIP call signaling messages. We piggyback our token submit protocol on top of the call setup as users will accept and reject the call based on the token. Token issue and transfer occur at the end of a successful call and are piggybacked on top of call teardown. For call setup, E.721[160] recommends an average delay of no more than 3.0, 5.0 or 8.0 s, for local, toll and international calls, with the 95th percentiles set at 6.0, 8.0 and 11.0 s, respectively. Looking at the token submit times from Figure 13, we see that other than for direct tokens that are submitted ($x = 0$), token submit times are greater than 3 s and increase by 1.5 s every hop. This implies that direct tokens offer acceptable call setup delays while tokens that have undergone one or two transfers will fall within the 95th percentile. Tokens that have undergone three transfers (four hops away) and beyond seem to have unacceptable call setup times.

We, however, note that for callers who are more than three hops away, this might be a fair penalty to pay to be introduced to a user. VoIP systems like Google Talk use a Turing Test for all users who are calling for the first time and successfuly completing the test takes more than the 9 s that our system requires for a user four hops away. In addition, this serves as a potential deterrent for malicious users who could eventually obtain tokens after a large number of hops. Furthermore, this is a one time cost, as after the introduction call, if interactions are favourable then the newly introduced user will start receiving direct tokens. Nonetheless, taking this and the UDP packet size limit into account, we only allow up to three transfers (users four hops away) in this implementation.

During call teardown after a successful call (based on a thresold call duration value), the caller can decide to either issue a token of his own or transfers another user's token. In this chapter, we use a simple strategy to make this decision. Specifically, when user A_1 calls user A_2, he issues tokens if A_2 has lesser than a threshold number of A_1's tokens, or if A_1 does not have sufficient number of tokens of any other user to transfer. In all other cases, A_1 transfers the token of a user from whom it has collected the maximum number of tokens.

To evaluate the combined system, we setup a simulation with 4 domains, each serviced by a proxy that handles 50 users, a total of 200 users in the system. In addition, we have a DNS server, a cryptography server and a statistics server. The DNS server translates domain names to the correct proxy IP address. The cryptography server generates the common parameters and doubles up as the CA. The statistics server calculates statistics including true positives, true negatives, false positives and false negatives. Initially each client requests the cryptography server for the common parameters and uses them to generate keys. It gets the keys certified by the CA and then it is ready to make and receive calls. The distribution with which it makes calls is dependent on the type of user the client represents. Clients can behave either as

an honest user or one of two types of spammers: (1) *engaging spammers* are able to engage users with a certain probability both when they receive calls and when they make calls, (2) *fleeting association spammers* are able to engage users only till the completion of some activity. Honest users makes calls to other phones with inter call and call duration values that are Poisson distributed. The choice of call recipient is Zipfian distributed. Spammers make calls to as many other users as possible. Honest users issue or transfer tokens based on a threshold call duration strategy. Spammers issue or transfer tokens to increase the number of spam calls. All spammers are inclined to collude with other spammers. In the simulation, 100 s of simulation time is equivalent to 1 day of real time. Each run lasts 20 days (2000 s).

Choice of Learning Period: The learning period is a duration of time just after a user is introduced into the system. During this time, the user accepts all calls to obtain a sizeable starting set of tokens from his SN. These tokens enable the user to call his SN and also disseminate his tokens so that others can call back. The learning period ensures that when a honest user is introduced into the system, he becomes selective about the calls he accepts only when he has a significant supply of tokens from members of his SN and his SN has a significant supply of his tokens. We assume that during this learning period spammers do not discover the user and therefore cannot spam the user. The graph in figure 15 shows the false positive rate (FPR) for 200 users, all honest, for different initial learning periods. The time axis starts 1 day into the running of the system as this is the minimum learning period that was used. The stabilized false positive rate shows an exponential drop with increasing learning periods. For learning periods of 1, 2 and 3 days it is $\approx 11\%$, $\approx 3\%$, $\approx 1.7\%$, respectively and thereafter stays around $\approx 1.5\%$ for higher learning periods. After learning periods of 3 days or more, users have a significant supply of tokens and can obtain tokens of users who are four hops away through the token transfer mechanism, resulting in a low false positive rate. Shue et. al.[136] studied the onset of spam

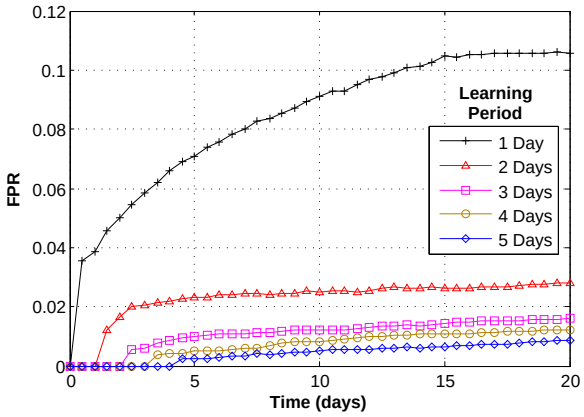


Figure 15: Learning Period - False Positive Rate

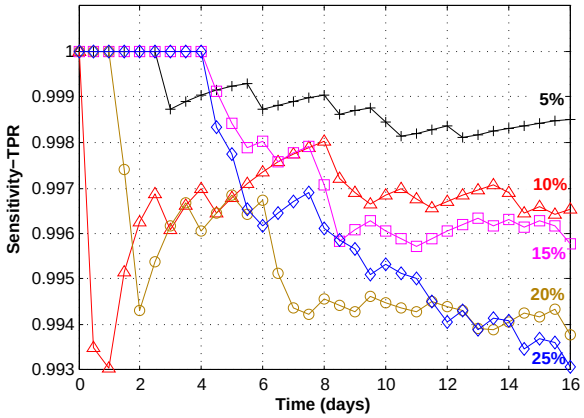


Figure 16: Engaging Spammers - True Positive Rate

and found that accounts that post their addresses on less popular websites will be discovered and receive spam only after 3 days. If we assume this holds for VoIP addresses too, then as long as users do not aggressively broadcast their addresses, a learning period of 3 days is feasible and provides a low enough FPR. In addition, since learning periods of more than 3 days do not reduce the FPR significantly, we use a 3 day learning period for the rest of the simulations.

Spammers that Engage Users In Conversation: In our system, users issue or transfer tokens only when a call lasts for more than a threshold duration. Spammers without the ability to engage users will never get tokens, even if they do manage to get users to inadvertently call them. However, spammers thrive because some honest users are fooled into believing the legitimacy of the spam content. To model this,

we associate with all users a value between 0 and 1 that represents the ability to engage another user in conversation. This value is set high for honest users and we configure spammers with various levels of engagability. Based on the engagability of the spammer, a user will inadvertently either issue or transfer tokens to spammers. Spammers can collude and therefore can collaboratively glean tokens. Spammers are introduced into the system immediately after the learning period (3 days). The results for different values of spammer engagability for a system with 20% spammers are shown in figure 16. For all cases our token framework is able to achieve a high sensitivity, of over 99%, in blocking spam calls. Spammers will find it hard to engage users in conversation and even when they do, the single use tokens only allows a limited number of calls. On the other hand, an honest user, due to his ability to carry on a conversation will first receive tokens of his immediate SN, and then receive tokens from his extended SN. From figure 16, small values of engagability (5%, 10% and 15%) result in a low false negative rate (FNR) and this rate stabilizes early in the run. However, spammers with a higher ability to engage users (25%), are able to make more calls at an ever increasing FNR, largely due to the collusions with other spammers. For spammers with 25% engagability, the final stabilized FNR was close to 1% and for 35% the FNR was close to 1.3%.

We also studied the effects of introducing 10% *fleeting association spammers* who behave adaptively. When just introduced to a user, they behave legitimately but soon start spamming the user. Spammers that behaved normally for periods of 1 and 2 days and then started spamming were able to achieve a FNR of $\approx 4.5\%$ and $\approx 17\%$. These values show that the success of a spammer increases significantly with the amount of time he is able to behave normally. However, the system reacts quickly and within a day reduces the FNR to under 10%.

4.7 Conclusion

In this chapter, we created a single use transferable token framework that captures interaction history in a privacy preserving manner by enhancing delegatable anonymous credentials. This allows us to prove the existence of a social network path without revealing the intermediate actors in the path. We show how we can use this to enhance CallRank by using it in a VoIP setting to prevent VoIP spam while being privacy preserving. We now broaden our scope and look at ways to create effective identities across the entire telecommunication landscape.

CHAPTER V

PINDR0P: USING SINGLE ENDED AUDIO FEATURES TO DETERMINE CALL PROVENANCE

The current telephony infrastructure allows users to communicate using a variety of technologies that pass through various providers within PSTN, cellular and VoIP networks. Each of these telecommunication networks adopt their own set of standards, from the underlying transport protocols to the codecs used to transmit media. Yet, they seamlessly interact through a variety of conversion mechanisms. A call may traverse multiple such networks, taking advantage of the benefits offered by each before reaching its final destination.

The diversification of telephony infrastructure significantly reduces the integrity associated with call metadata, such as Caller-ID [2], as it is either not transferred across these networks or is transferred without verification. This allows easy manipulation of metadata by hardware and software including soft phones on desktop computers. For example, between January 21st and 26th of 2010, customers of banks in four states received calls asking them to reveal personal information including credit card and PIN details. Many of these attacks use VoIP phones to anonymously and inexpensively dial a large number of customers while forging the Caller-IDs of these banks [104].

In this chapter, we develop PinDr0p[1], an infrastructure to assist users in determining the provenance of a call — the source and the path taken by a call. Through a combination of signal processing and machine learning, we show that regardless of

[1]Our mechanisms take advantage of audio and path artifacts that, like the sound made by the drop of a pin, are largely inaudible to the human ear.

the claimed source, the audio delivered to the receiver exhibits measurable features of the networks through which the call was delivered. For example, calls that traverse a VoIP network experience packet loss that results in perceivable effects in the final call audio. Such artifacts are noticeably absent in calls that have only traversed cellular or Public Switched Telephone Networks (PSTNs). In particular, the codec transformations applied by multiple intermediary PSTNs, VoIP and cellular networks, in combination with packet loss and noise characteristics, allow us to develop profiles for various call sources based solely on features extracted from the received audio. In the absence of any verifiable metadata, these features offer a means of developing source fingerprints that help compare and distinguish different incoming calls.

We make the following contributions:

- **Identify robust source and network path artifacts extracted purely from the received call audio:** We show that the received call audio provides extractable features that are strong identifiers of the networks that the call has traversed, allowing us to determine the provenance of a call. These include degradations (packet loss in VoIP) and noise characteristics of codecs unique to each network.

- **Develop call provenance classifier architecture:** We develop a multi-label machine learning classifier based on the extracted features to correctly identify the provenance of an incoming call with 91.6% accuracy with as little as 15 seconds of audio. Because PinDr0p does not rely on metadata available in some networks (e.g., VoIP) or cryptography, it is more readily deployable across the diverse devices and networks that make up modern telephony systems.

- **Demonstrate our robustness in identifying call provenance for *live* calls:** We make calls using PSTN phones, cellular phones, Vonage, Skype and other soft phones from locations across the world and are able to distinguish

between them with 90% accuracy with only a small sample being labeled. As we increase the number of such labels we are able to distinguish between these calls with 100% accuracy. This demonstrates that PinDr0p makes VoIP-based phishing attacks harder and provides an important first step towards a Caller-ID alternative.

We note that while our approach does not provide the same guarantees as the use of end-to-end cryptography, it is also not encumbered with the difficulties of key distribution, management and the requirement that both endpoints are capable of such operations. The guarantees provided by our approach are instead more akin to traceback techniques from IP networks [131]. However, PinDr0p does not mandate the modification of the core infrastructure to attach additional metadata in-transit as our provenance information is extracted directly from the received audio. While adversaries may attempt to modify their attack in order to circumvent PinDr0p (e.g., change codecs, replicate the noise profile *and* change the physical location from which an attack is launched to match packet loss characteristics), our approach significantly increases the difficulty of successfully launching such an attack and improves the chances of identifying an attacker.

The remainder of this chapter is organized as follows: Section 5.1 discusses the details of our proposed call provenance mechanism; Section 5.2 details our experimental setup and results; Section 5.3 presents experimental results from a real-world attack scenario; Section 5.4 offers further insight into our scheme and discusses trade-offs and limitations;

5.1 Call Provenance

The provenance of a call describes the characteristics of the source and traversed networks. This information can be used to create fingerprints that help distinguish and compare different calls in the absence of verifiable end to end metadata. For example,

provenance can be used to identify if a call has passed through a VoIP network and, if it has not typically done so, alert the receiver of the change. At the very least, provenance must be able to distinguish between traffic that has traversed different telephony networks: PSTN, cellular and VoIP. We investigate whether this can be achieved with only the audio content available at the receiving end of a call. This approach is attractive as provenance can be determined without access or modification to intermediate network elements such as gateways or routers.

As a call traverses multiple networks, the audio is repeatedly re-encoded with the network's choice of codec. To illustrate, a Skype call to a landline is initially encoded using G.729 and re-encoded using G.711 when it encounters the VoIP-PSTN gateway. If we can extract artifacts of each of the applied codecs from the received audio then simple codec to network translation (G.729 $\Longrightarrow$ VoIP) determines call provenance. In addition, identifying the codec used in a particular network helps characterize that network. However, codecs like G.711 are widely used in both PSTN and VoIP systems, implying codec detection alone is insufficient. Therefore, we seek additional differentiators.

Networks themselves introduce degradations into call audio. In VoIP, there are packet losses which are not seen in circuit switched PSTN networks. Similarly, mobile phones have bit errors due to fading effects on radio channels. The loss of an entire packet containing 20 ms of speech is measurably different from a small number of incorrect bits. These features are more robust than simply extracting codec information as packet loss and bit errors are hard for an adversary to control — *an adversary bounded by a lossy connection, many miles away, cannot spoof a lossless, dedicated PSTN line to a bank.*

Solution Overview: To identify and characterize the different networks a call has traversed, we focus on degradations specific to each network. We first demonstrate how we can identify and characterize a VoIP network by detecting packet loss

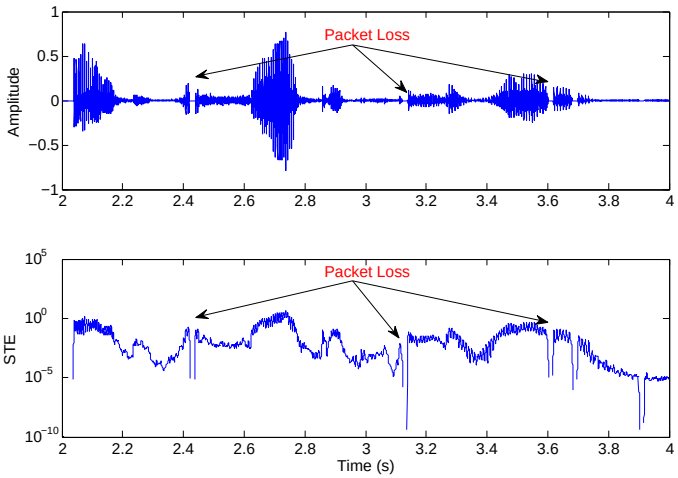


Figure 17: Packet Loss and Corresponding Energy Drop. The breaks in the signal (top) that occur due to packet loss are more accurately determined using the short time energy (bottom) of the signal.

or concealed packet loss in the received audio. We then show how PSTN and cellular networks can be identified and characterized due to their vastly different noise characteristics. Finally, since the quality of the received audio significantly degrades with the number of networks traversed, we extract quality specific features. We create a feature vector that aggregates feature values obtained from the packet loss, noise and quality measurements and use it to train a multi-label classifier to identify the networks that a call originated and traversed. In addition, we demonstrate how the feature vector provides call provenance fingerprints that can be used to consistently identify a call source.

5.1.1 Identifying VoIP Networks

5.1.1.1 Detecting Packet Loss

Within an IP network a lost packet can be easily identified using the sequence numbers present in each packet (metadata). However, these sequence numbers are lost once the call is retransmitted over another telephony network. Accordingly, we must identify artifacts of these lost packets from the received audio. The top graph in Figure 17 shows two seconds of speech encoded with G.711 and transmitted through a VoIP

network with a packet loss rate of 5%. The effect of a lost packet is sometimes visibly identifiable by a break in the waveform (annotated by arrows). However, such loss can be detected more accurately by determining the short-time average energy of the signal, as shown in the bottom graph in Figure 17.

Short-time average energy (STE) is traditionally used in speech analysis to detect words surrounded by pauses as they cause abrupt drops in energy levels. This can be adapted to detect a packet loss, which also causes an abrupt decrease in energy. STE for a signal $y(n)$ is defined as:

$$E_n = \sum_{m=-\infty}^{\infty} y^2(m) \cdot w(n-m),$$

where E_n is the STE for a window of speech $w(n)$. Specifically, $w(n)$ is a sliding Hamming window of length N, where the speech samples closer to n are weighted more than those at the window edge. For the codecs we consider, a packet contains at least 10 ms of audio represented by 80 samples of speech. By making our window length less than 80, multiple values of E_n are completely influenced by a dropped packet. This results in the breaks in energy shown in Figure 17. We detect packet loss by looking for a significant drop in energy followed by an energy floor, accompanied by a significant energy rise.

We note that the presence of all three of these characteristics is necessary to detect packet loss as each appears individually even in speech that has not experienced any packet loss. For instance, in Figure 17, we see a significant rise in energy at approximately 2 seconds due to the start of a speech segment. This is a result of Voice Activity Detection (VAD) in VoIP systems where packets are only sent during active speech to reduce bandwidth. Similarly, when a speech segment ends there is a significant drop in energy. Figure 18 shows the STE of a 15 second speech sample, encoded with G.711 and transmitted through a network with 5% packet loss. The dots at the bottom are the actual packet losses and the ones above are the packet

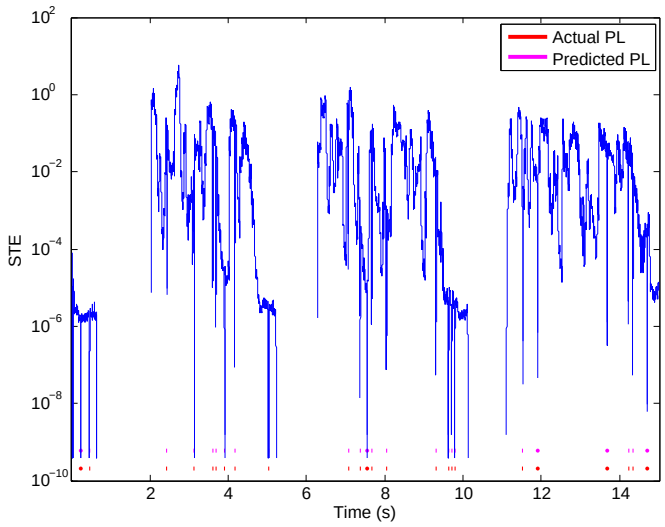


Figure 18: Packet Loss Prediction. The dots below show the actual losses and the ones above are identified by our algorithm. The close correspondence between the two indicates that we detect lost packets accurately.

losses identified by our detection mechanism. The close correspondence between the two shows that our detection mechanism identifies packet loss accurately.

Figure 19 shows false positive and false negative cases for our detection mechanism. In the top graph, a packet loss occurs at the start of a speech segment (7 seconds). Since we classify packet losses based on an energy drop, floor and rise, such losses are not detected. Note that this conservative approach reduces our false negatives at the cost of potentially missing a small number of losses at the beginning and end of speech. False negatives are shown in the bottom graph in Figure 19 at 3.2 seconds and occur in the rare case when speech stops and starts in quick succession, with the stop duration corresponding to a multiple of 80. This pattern occurs only when there is a voiced "plosive," or a stop sound in speech, such as the *b* sound in the word "*about*."

Each time a packet loss is detected, the length of the energy floor also reveals the codec used in a particular VoIP network. Figure 20 shows the effect of packet loss on two VoIP networks using different codecs: iLBC which encodes 30ms and Speex which encodes 20 ms of speech per packet. The length of the energy floor is larger

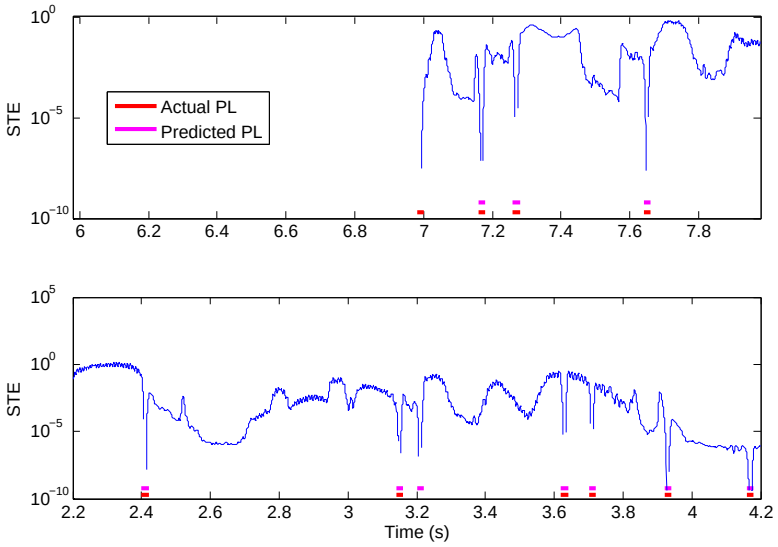


Figure 19: Scenarios showing a false negative (top at 7 seconds) and a false positive (bottom at 3.2 seconds).

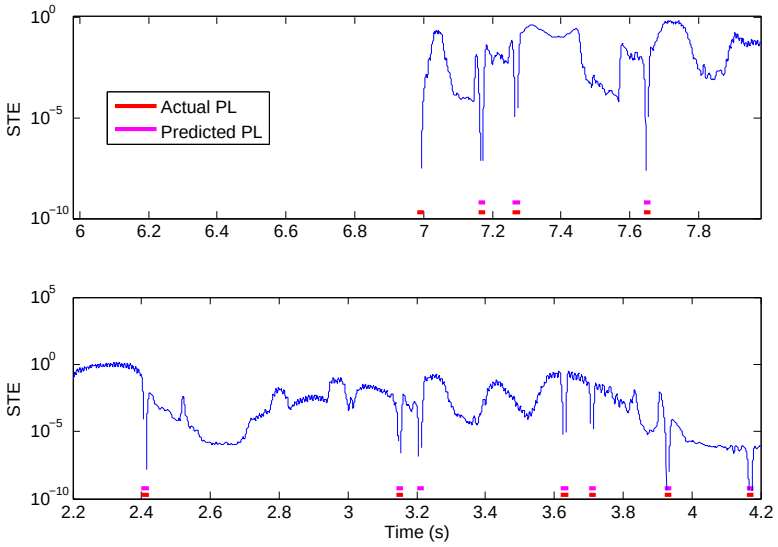


Figure 20: Packet loss affect codecs differently. iLBC encodes 30 ms of audio per packet and therefore a packet loss results in more audio lost in comparison to Speex which encodes 20 ms of audio.

for iLBC than Speex. In addition, since G.729 encodes 10 ms and G.711 encoded 20 ms per packet by default, the length of the energy floor is a good indication of the codec used. We might identify the wrong codec when consecutive packets are dropped as two consecutive packets dropped in a network using G.729 (10 ms audio) will be similar to a single packet dropped in a network using G.711 (20 ms audio). However, the probability of consecutive packets being dropped is lower that the probability of a single dropped packet and we can identify the codec based on the most commonly occurring energy floor length.

To summarize, short time energy provides a highly accurate mechanism to determine packet losses and the detection mechanism can also be used to identify the codec used. Therefore, when a call traverses a potentially lossy VoIP network, the packet loss rate and the codec used in that network can be extracted from the received audio.

5.1.1.2 Detecting Concealed Packet Loss

Some VoIP systems employ packet loss concealment (PLC) algorithms to prevent short speech gaps from affecting call quality. Such concealment can be carried out at the receiver (reactive) or with the assistance of the sender (proactive). In reactive recovery, the lost packet is concealed either with silence, noise or is regenerated by interpolating previously received packets. Proactive recovery algorithms include redundant information such as the previous packet's audio with each packet. This approach incurs a bandwidth overhead and is rarely used. We focus on identifying the effects of receiver side recovery algorithms on the audio and leave sender side algorithms to future work.

When the concealment mechanism is silence or noise substitution, the STE-based algorithm from the previous section can be used to detect packet losses by suitably adjusting the energy floor to correspond to the noise floor. Most VoIP codecs, however, reconstruct lost packets from previous packets. G.711 uses waveform substitution

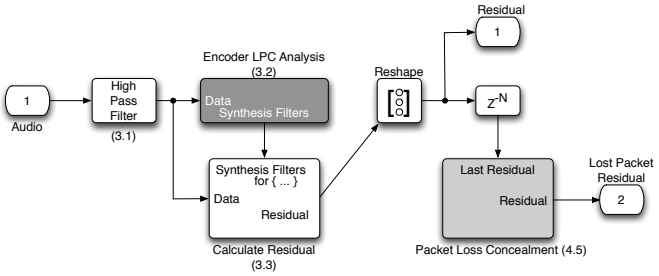


Figure 21: The iLBC packet loss concealment detection algorithm. Because lost packets are regenerated in a largely deterministic fashion from the residual and synthesis filters of the previous packet, such packets can be detected by measuring the correlation between the residuals of sequential packets.

to repeat a portion of the previous packet [151]. In codecs designed specifically for VoIP such as iLBC or Speex, the concealment algorithm is more elaborate in order to improve robustness to high packet loss rates. Fortunately, we observe that concealment techniques are predominantly deterministic and a detection mechanism can be created that exploits the correlation between reconstructed packets and previous packets. We discuss the details of the PLC algorithm in iLBC to provide further clarity.

iLBC uses a linear predictive coding (LPC) algorithm to represent speech in a significantly compressed form. LPC is based on the source filter model of speech production, where the larynx (source) produces sound energy, which when voiced consists of a fundamental frequency (pitch) and its harmonics. This sound energy is then shaped (synthesis filters) by the vocal tract (throat and mouth) into enhanced frequency bands known as formants, which provide speech its intonation. The LPC algorithm inverse-filters the formants from the speech signal to leave behind the original sound energy, known as the residual. A codec like iLBC uses the residual, the synthesis filters and dynamic codebook encoding to reduce the original speech into a set of parameters which can be transmitted. The decoder uses these parameters to reconstruct the residual and the synthesis filters which when combined re-synthesize the speech. When a packet is lost, the decoder uses the residual from the previous

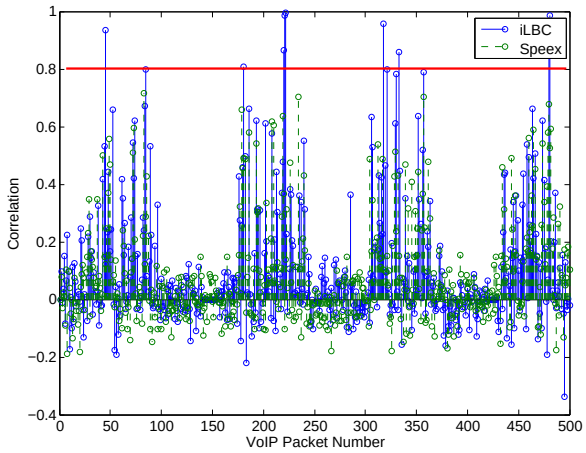


Figure 22: The result of testing for the presence of highly correlated in-sequence packets based on the iLBC packet loss concealment algorithm. The algorithm specifically detects iLBC (solid blue lines) while remaining agnostic to other codecs such as Speex (dotted green lines)

packet and creates a new pitch synchronous residual for the packet to be concealed. Additionally, a random excitation is added to the new residual (non-deterministic part). The new residual along with the synthesis filters from the previous packet are used to create speech that will be substituted for the lost packet. Therefore the new residual will be strongly correlated to the previous packet's residual. To detect PLC in iLBC we first split the received audio into packets containing 30 ms audio each (the default for iLBC's). We then create a pitch synchronous residual from each packet and compare it to the residual extracted from the next packet. As these quantities are generally not highly correlated, the detection of an association between sequential packets is a very strong indicator of iLBC's packet loss concealment algorithm. The packet loss concealment algorithms for the other codecs, though different, can be detected based on how sequential packets are correlated.

Figure 21 shows a detailed block diagram for the iLBC PLC detection algorithm. Since the encoding procedure in iLBC already extracts the residual from the audio, we first split the audio into 30 ms chunks and apply the encoding steps defined in Section 3.1 to 3.3 of iLBC RFC 2951 [75]. This includes running a high pass filter

to remove noise in the audio, performing LPC analysis to extract the synthesis filters and then using the synthesis filters along with the data to extract the residual, r. We use r to generate a pitch synchronous residual r' as defined in Section 4.5 of iLBC RFC 2951. r' will be strongly correlated to the residual from the next chunk of 30 ms of audio if that packet had been lost. We calculate r and r' for each chunk and report high correlations between as indications of PLC.

Figure 22 shows the correlation between residuals of a 15 second speech sample encoded with the iLBC codec (solid blue lines) and transmitted through a VoIP network with a loss rate of 10%. At each high correlation point (above 0.8) we confirm from our logs that the particular packet was lost. To show that the PLC detection algorithm is specific to iLBC, we run it on the same 15 second speech sample encoded with Speex instead and transmitted through the 10% loss rate VoIP network. The results are again shown in Figure 22 as the dashed green lines. Though packets were lost in this case too, the detection algorithm does not show high correlation between residuals, confirming that we can create PLC detection algorithms specific to the way each codec conceals packets. Since all the codecs use different concealment strategies, in addition to detecting concealed packet losses our algorithms also provide a strong indication of the codec used in a particular VoIP network.

Finally, in Figure 22 we observe that for the 15 second sample encoded with iLBC, 54 out of the 501 packets (loss rate $= \frac{52}{501} = 10.38\%$) were lost and we are only able to identify 9 correlations. This is largely due to the fact that the PLC algorithm is not completely deterministic (random excitation). However, the number of concealed packets detected is still indicative of the loss rate. To show this, we ran our detection algorithm over 15 seconds of 20 male and female American English speech samples from the Open Speech Repository [164] encoded with iLBC and transmitted through VoIP networks with 0, 1, 5 and 10% loss rates. The association between the number of concealed packets detected and the packet loss rate are shown in Figure 23. It shows

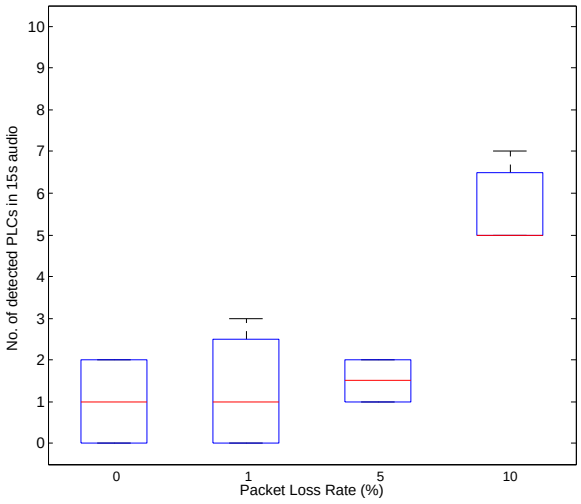


Figure 23: Number of concealed packets detected with increasing loss rate in a 15s speech sample. The median number of concealed packets detected by our algorithm increases with increasing loss rate.

the median and the 25th and 75th percentiles with whiskers specified as .5 times the interquartile range. We see that the median number of concealed packets increases significantly as the loss rates increase. Therefore, the PLC detection algorithm can make approximations of the loss rate but is not as accurate as the detection algorithm for unconcealed packet losses.

Our packet loss and packet loss concealment detection algorithms identify three aspects about the provenance of a call: (1) Whether the call traversed a VoIP network, (2) the packet loss rate in that network and (3) the codec used in that network. (1) identifies if there are VoIP networks in the path of a call and (2) and (3) characterize the VoIP network.

5.1.2 Identifying PSTN and Cellular Networks Through Noise Profiling

Now that we are able to identify and characterize VoIP networks, we can look for codec specific artifacts in the received audio to identify PSTN and cellular networks.

Waveform codecs like G.711 are used mostly in PSTN networks as they capture speech without any compression and require much higher bandwidth (64 kbps) than most other codecs. They tend to introduce noise only during speech activity resulting

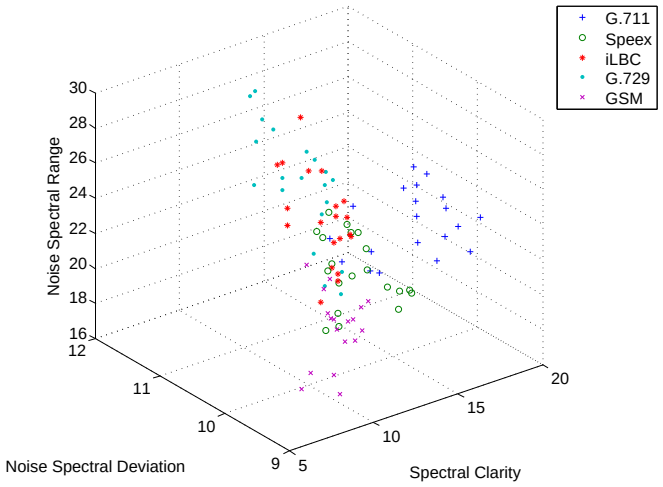


Figure 24: The noise profile of G.711 is significantly different from other codecs, allowing us to identify it when it is used in a network.

in a strong correlation between the noise and the signal. This is known as multiplicative noise and its presence can be determined based on spectral statistic metrics: spectral level range and the spectral level deviation. Furthermore, the spectral clarity for such a codec, or the measured crispness of the audio, is very high. In contrast, since cellular networks require efficient use of bandwidth they use high compression codecs like GSM-FR (13 kbps). The spectral clarity of such codecs suffer due to the significant compression. Spectral clarity quantifies the perceptible difference in call quality that we experience when talking on a landline versus a mobile phone. Figure 24 shows the spectral clarity, the spectral level range and deviation for 20 male and female American English speech samples from the Open Speech Repository [164] encoded and decoded using the different codecs. We see that G.711 and GSM-FR can be clearly identified. Once we identify the codec using these metrics we can do a simple codec to network translation to determine if a call has traversed a PSTN network or has originated from a cellular network. Furthermore these three metrics provide a noise profile of the network thereby characterizing it.

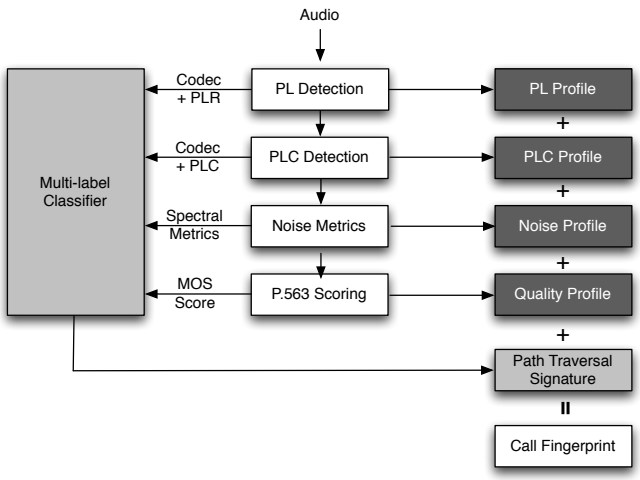


Figure 25: The PinDr0p call provenance extraction algorithm. After the applied codecs have been detected, packet loss rates are compared against individual source profiles. The resulting signature can be used to judge the provenance of an incoming call.

5.1.3 Extracting Provenance Data

We have seen how packet loss and packet loss concealment detection identifies and characterizes any traversed VoIP network. Similarly, the noise profiles identify and characterize any PSTN and cellular network. Together, we can create fingerprints that detail the provenance of a call.

Call provenance fingerprints consist of two parts: (1) the path traversal signature and (2) detailed characterization of each network in the path traversal signature. The path traversal signature identifies the networks that a call traversed and the codec used. The characterization provides more details of each network. The features we extract can be used towards both these parts as shown in Figure 25. To obtain the path traversal signature we first train a multi-label classifier as shown in Figure 25 using a repository of speech samples. Each sample is subjected to codec transformations and network degradations depending on the networks it traverses (details in Section 5.2). For each of the resulting audio samples, we first look for packet losses. If present, we calculate the packet loss rate which forms the packet loss profile and then add the extracted codec information and the rate (as G.711 with some loss rate

Table 4: Call Traversal Scenarios.

Configuration	Scenario	# Simulated Samples
Single Network Traversal		
PSTN - PSTN	Plain old telephone call	20
Mobile - Mobile	Short distance call b/w cell phones	20
VoIP - VoIP	Unfederated call b/w VoIP clients e.g., Google Talk	60
Two Network Traversal		
PSTN - Mobile	Call b/w PSTN landline and cell phone	320
PSTN - VoIP	Call b/w PSTN landline and VoIP client e.g., Skype-Out	360
Mobile - VoIP	Call b/w cell phone and VoIP client	560
Three Network Traversal		
PSTN - VoIP - Mobile	International call using calling cards	1200
PSTN - VoIP - PSTN	Same as above	240
Mobile - VoIP - Mobile	VoIP call bridging b/w two mobile phones e.g., Google Talk	960
Mobile - PSTN - VoIP	Call b/w mobile using a PSTN core network and a VoIP client	400
Mobile - PSTN - Mobile	Similar as above	80
VoIP - PSTN - VoIP	Call b/w two commercial VoIP clients e.g., typical Vonage call	720
		Total = 4940

indicates a VoIP network) to the feature vector. Next, we apply the correlation algorithm to detect packet loss concealment. If the correlation algorithm finds concealed losses, the corresponding codec is again added to the feature vector along with the number of concealed packets (PLC profile). We then extract the noise profile for the call audio and add the spectral metrics to the feature vector. Since the quality of speech degrades with the number of networks traversed we also obtain call quality metrics from a single ended quality tool, P.563 [152] and add this to the feature vector. The multi-label classifier is then trained on each sample's feature vector and label. A sample has five labels, each indicating the presence or absence of a codec. For example a speech sample in our repository that was encoded using GSM-FR (originated at a cellphone), then re-encoded using iLBC (traversed a VoIP network) and finally re-encoded using G.711 (receiving end point is a landline) would have a '1' for three labels (GSM-FR, iLBC and G.711) and a '0' for Speex and G.729. Multi-label classifiers have been used significantly in text categorization [157, 108, 175] and we use a set of standard reduction techniques to convert the multi-label data into a single label model. The classifier then learns which features best predict the presence or absence of a label.

For any new call audio we perform the same procedure, but do not add any label as the classifier will predict a set of labels based on the learned model. The prediction of the classifier for the path traversal signature, along with packet loss, noise and quality profiles, represents the call provenance fingerprint for a particular source in PinDr0p.

5.1.4 Security Implications

The path traversal signature and the complete provenance fingerprint provide a useful security framework in the absence of any verifiable metadata. The traversal signature alone can be used against adversaries who are bound by operating constraints. For example, adversaries trying to spoof a dedicated line to the bank might use VoIP due to the fact that they can remain largely anonymous and can make a large number of inexpensive calls. However, the path traversal signatures for these two calls will be different. To address this, the adversary can switch to a landline, in which case he has lost the ability to easily make a large number of calls and potentially compromised his anonymity.

We can also use the complete provenance fingerprint against adversaries as it also characterizes individual networks. Since this involves capturing detailed profiles of these networks traversed, an adversary trying to spoof a call needs to be able to match all these profiles. We show in Section 5.3 that our fingerprints are able to discriminate between sources that are in the same city using the same provider, demonstrating that matching an entire fingerprint is extremely difficult. Accordingly, we believe our approach is a significant first step in creating suitable defenses against a host of attacks possible in today's diverse telephony infrastructure.

5.2 Evaluation

We evaluate our approach based on two metrics: (1) the accuracy of our multi-label classifier in predicting the correct network traversal signature of a call and (2) the

ability of our provenance fingerprint to consistently identify a call source. We discuss the evaluation of the first in this section and analyze the second in the following section.

5.2.1 Experimental Setup

We train and test the multi-label classifier against a repository of speech samples that are subjected to a representative set of real-world call traversal scenarios and network degradations. We assume calls can traverse one, two or three networks as most call scenarios fall into one of these cases; however, our methodology can be extended to deal with additional transcoding. Table 4 shows the considered call traversal configurations. Single network traversals represent calls that are contained within one system. For example, the VoIP-VoIP scenario occurs when two Skype users call each other. Since both clients are connected to the Internet, they communicate through a set of relays (supernodes) and the call stays completely within the IP network. Two network traversals are calls from users on one telephony technology to users on another. There are six possible combinations and for brevity we only list three of them, in each case subsuming the symmetric traversal scenario (i.e., PSTN-Mobile and Mobile-PSTN are categorized as a single scenario). Finally, three network traversals occur when providers attempt to take advantage of the benefits offered by each telephony technology. For instance, while calls between two Vonage clients within the US can be completely VoIP-VoIP, Vonage specifically transmits the call over the PSTN backbone due to its QoS guarantees. Similarly, most international calling card services use VoIP across the Internet as this provides an inexpensive calling alternative.

Our experiments use speech samples from the Open Speech Repository [164], which contains samples of 20 different American English speakers, 10 male and 10 female, speaking phrases from the Harvard sentence list [13]. These samples are used

for standardized testing of PSTN, VoIP and cellular systems as recommended by the IEEE Recommended Practices for Speech Quality Measurements [20]. Each sample is 40 seconds long, but we consider only the first 15 seconds, as call quality algorithms such as P.563 typically use this length to determine call quality metrics.

We consider the most popular narrowband codecs for encoding calls in our experiments. Specifically, we use G.711 for PSTN systems, G.711, G.729, iLBC and Speex for VoIP systems, and GSM for cellular systems. Calls traversing two telephony networks (e.g., VoIP to cellular) are transcoded to the new codec.[2] Since transcoding is not always defined for a pair of codecs, we follow the common practice of converting to and from an intermediate G.711 form. We use the PJSIP [117] suite of applications to encode and perform the necessary conversions between codecs. PJSIP contains open source SIP and media stacks and is part of the European Broadcasting Union Audio over IP standard [66]. It supports G.711, iLBC, Speex and GSM. For G.729, we integrate the Intel Integrated Performance Primitives Library [86] into PJSIP.

In addition to the codecs, each traversed network is characterized by its signal degradation characteristics. VoIP networks experience packet losses which typically increase in correlation with factors such as routing distances, "last-mile" unreliability, network congestion and over-subscription. For VoIP networks, we simulate packet loss rates of 1, 5 and 10%. For bit errors occurring from multi-path fading radio channels in mobile networks, we use a GSM traffic channel simulator developed for Simulink [107].

Experiments are conducted by taking one speech sample from the Open Speech Repository and encoding it with the appropriate codec using PJSIP. Samples corresponding to packet losses or signal degradations found in the traversed telephony network are also generated and tested (e.g., packet loss in iLBC, multi-path fading

[2]Recall that VoIP calls can cross multiple autonomous systems throughout the Internet without being transcoded.

Table 5: Accuracy of multi-label classifier using C 4.5 decision trees.

Metric	Definition	BR	LP	RAkEL
Hamming Loss	$\frac{1}{\|D\|} \cdot \sum_{i=1}^{\|D\|} \frac{\|Y_i \triangle P_i\|}{\|L\|}$	.09	.1	.05
Accuracy	$\frac{1}{\|D\|} \cdot \sum_{i=1}^{\|D\|} \frac{\|Y_i \cap P_i\|}{\|Y_i \cup P_i\|}$	83.7%	83.7%	91.6%
Precision	$\frac{1}{\|D\|} \cdot \sum_{i=1}^{\|D\|} \frac{\|Y_i \cap P_i\|}{\|P_i\|}$	91.5%	89.3%	93.7%
Recall	$\frac{1}{\|D\|} \cdot \sum_{i=1}^{\|D\|} \frac{\|Y_i \cap P_i\|}{\|Y_i\|}$	90.3%	89.3%	97%

in GSM). We also append the codec multi-label for each generated sample. We aggregate all possible resulting speech samples into a corpus. The number of samples for each of the traversal scenarios is shown in Table 4.

We run the feature extraction algorithms described in Section 5.1.3 on each of the speech samples and then train and test a multi-label classifier on the resulting feature vector and label. We use Mulan [153], an open source Java library for multi-label learning, to create our machine learning classifier.

5.2.2 Classification Results

Multi-label classifiers can use a variety of reduction techniques including Binary Relevance (BR), Label Power (LP) set and Random k-Labelsets (RAkEL) [158] to convert the multi-label into a single label. The resulting labels can then be classified by any of the traditional single-label classifiers. We use C4.5 decision trees as the underlying single-label classifier as it outperforms other classifiers that we considered including Naive Bayes and Neural Networks. Using the corpus described above, we use 10-fold cross validation to measure the accuracy of the multi-label classifier under the three reduction techniques. Our results are described in Table 5. We define the metrics as specified in the multi-label classification literature [157]. Let the multi-label dataset consist of $|L|$ labels (five in our case) and $|D|$ instances in the test set, with each instance i represented by feature vector f_i and label Y_i. The classifier C makes label predictions $P_i = C(f_i)$ for each instance f_i. For a test instance with known path

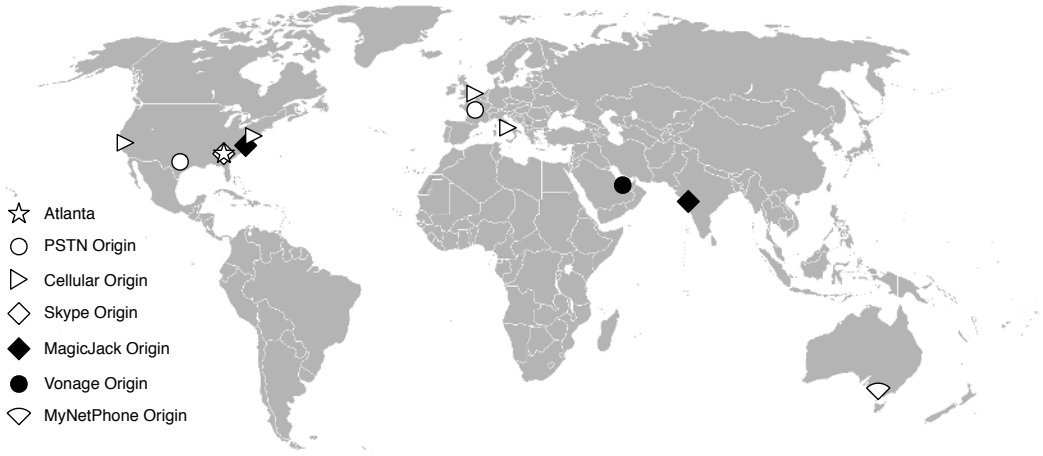


Figure 26: We tested our system using multiple sources from four continents: North America, Europe, Asia and Australia. Specifically, we recorded incoming calls from five different PSTN phones in Atlanta, GA, Dallas, TX, and France; four different mobile phones in Atlanta, GA, New York City, NY, San Jose, CA and London, UK; six VoIP phones in Atlanta, GA (Skype and Vonage), Baltimore, MD(MajicJack), Pune, India(MagicJack), Dubai, UAE(Vonage) and Melbourne, Australia (MyNetPhone).

traversal signature, the classifier predicts a label using only the feature vector. The metrics defined help quantify the difference between the predicted and actual labels.

We find that RAkEL has the lowest Hamming loss and the highest accuracy of 91.6%. The results show that we are able to predict which networks a call traversed with high accuracy. We also find that the majority of misclassifications occur for samples that traversed a VoIP network with 0% packet loss rate.

5.3 Real-World Testing

The complete provenance fingerprint of a call consists of the path traversal signature, and profiles for packet loss, concealment, noise and quality. If this fingerprint remains consistent for a call source, it provides valuable metadata that can be used to identify and distinguish different calls purely from the received audio. We asked different users to make a set of 10 live calls to our testbed in Atlanta, GA from 16 different locations around the world, including Australia, India, United Arab Emirates, United Kingdom and France. The complete list of locations is shown in Figure 26.

Each call lasts approximately 20 seconds. We extract features and profiles from

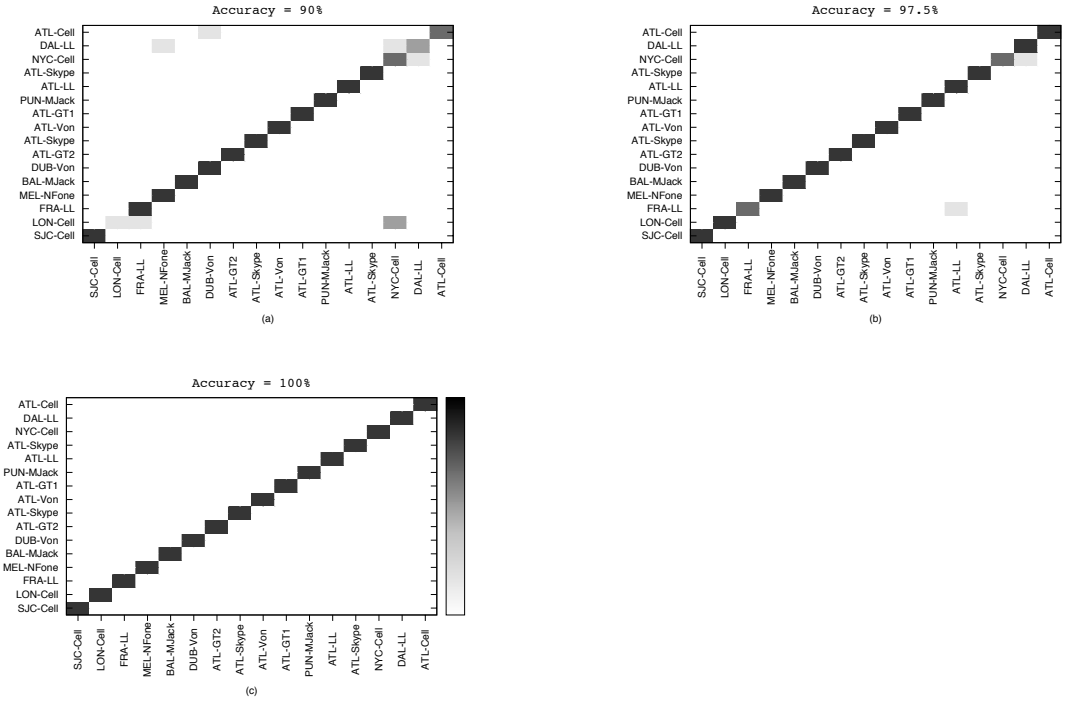


Figure 27: The confusion matrix for the live-captured call data trained with labels for (a) one set of calls, (b) three sets of calls and (c) five sets of calls from all call sources. The accuracy on even a singly labeled training set is 90% and quickly jumps to 100% with 5 labeled training sets.

the received audio and then label all calls from a call source with the same unique label. We then train a neural network classifier for N sets of the 10 call sets (set = one call from each source). We vary N from one to five and then test with five new call sets. This represents the scenario that a user labels a set of calls and expects subsequent calls coming from the same source to be labeled correctly by our algorithm. Our experiment evaluates the tradeoff between labeling effort and accuracy.

The results show that even if a single set of 16 calls is labeled, the remaining five sets of calls from the 16 different locations are identified with the correct call source label with 90% accuracy. The accuracy increases quickly to 96.25% for two, 97.5% for three, 97.5% for four and 100% for five labeled sets. Figure 27 shows the confusion matrix for 1, 3 and 5 training sets.

Even with a singly labeled training set (Fig 27(a)) we find that all VoIP calls

are correctly identified as they are easily distinguishable from the other networks. They are also distinguishable among themselves as they are geographically spread across Atlanta, Maryland, Dubai, Pune and Australia and each has a different packet loss concealment profile. In some cases we were pleasantly surprised by the actual differentiator. We found that Vonage calls from Atlanta were distinguishable from all the other VoIP calls based on its high spectral level range (noise profile) rather than the packet loss profile. We suspect that this is due to the fact that Vonage calls almost immediately transfer to the PSTN backbone for quality of service, while other services predominantly use VoIP. However, we did not observe this in the international Vonage call from Dubai where the call path would be predominantly VoIP, instead, to make the call affordable.

Figure 27(a) shows that even with a singly labeled training set we are able to distinguish between the three landlines from Atlanta, including the two from within the Georgia Tech campus, demonstrating that even for similar call sources the characteristics can be significantly different. We also see that three of the five calls from the London mobile phone are misclassified as a mobile phone call from New York and one call was misclassified as a landline call from France. The provenance of the call from London seems to be misclassified based on either the distance similarity (both coming from Europe) or the same origin network (cell). The number of misclassifications for the test set containing 80 calls (16 locations $\times 5$) drops significantly from 10 to 3 and then to 2 with increasing the number of training sets. With five labeled call sets being trained we have no misclassification showing that with each extra label the classifier becomes increasingly accurate.

The profiles that we capture for each source are consistent for the same call source but have enough variability to allow us to distinguish different call sources. Although we still require 15 seconds of call audio before being able to identify the provenance, we believe that an attempt to steal sensitive information (e.g., bank account numbers)

from a potential victim requires significantly more time. Accordingly, users should be sure to wait at least this amount of time before disclosing such information. We plan to investigate the uniqueness of a larger number of call sources as part of future work.

5.4 Discussion

In this section, we investigate some of the limitations of our current infrastructure and discuss a number of future extensions that will both improve the accuracy of our detection and its resistance to more active adversaries.

5.4.1 Limitations

Our call provenance infrastructure is designed to detect codecs and path characteristics associated with a given source. In spite of its relative strength, there exist a number of limitations associated with our current system. For instance, unlike Caller-ID systems, our call provenance infrastructure requires that the receiver answer the call before its source can be verified. This may not be useful to those using Caller-ID as a means of deciding whether or not to take a call. This shortcoming could potentially be addressed by pushing our mechanism into the cloud. Incoming calls could potentially be forced to first interact with a recording, which could collect sufficient audio for analysis, before reaching the intended target.

We currently rely heavily on packet loss characteristics of the path between sources and our testbed to differentiate VoIP fingerprints. While instantaneous packet loss rates certainly fluctuate, paths and their corresponding loss patterns are relatively stable in the Internet [114]. However, we recognize that our packet loss profiles may need to be more accepting of diurnal cycles and temporary anomalies and plan to study such issues in the future.

As an implementation decision, we currently associate a source with a single fingerprint. This assumption is appropriate when dealing with an immobile source such

as a corporate calling center. However, individual users may take advantage of the mobility allowed by VoIP software such as Skype to legitimately place calls from a number of different locations. The advantage in such a scenario is that the receiver is likely to recognize the caller's voice and can therefore manually associate new fingerprints to a particular source.

Lastly, we have attempted to analyze the most widely used codecs in our study. However, other less widely use codecs were not considered in this initial study. For instance, the Adaptive Multi-Rate (AMR) codec, which provides higher audio quality and is beginning to compete with GSM on mobile devices, and a handful of others such as the Enhanced Variable Rate Codec (EVRC) for CDMA networks will be considered as part of our future work.

5.4.2 Additional Applications

We have focused the work in this chapter on using call provenance to address Caller-ID spoofing attacks. However, the utility of PinDr0p is not limited to this task. While stories of VoIP-based phishing (vishing) have become popular in the media [168, 144], the extent to which such calls are occurring compared to traditional telephony fraud is unknown. The deployment of our infrastructure in a distributed fashion may help to answer this question. In particular, the use of call provenance in this space can assist in determining the prevalence and potentially the identity of individual vishing campaigns. While we leave the details of such an infrastructure to future work, we hope to be able to provide the security community with a tool for better understanding such attacks.

PinDr0p may also be useful as a means of authenticating channels. For instance, credit card and home security companies often use Caller-ID information as a second factor of authentication when customers call with account questions. Such organizations could increase the number and difficulty of questions asked of the caller based

on the measured provenance of the incoming call. In multi-factor authentication analysis, PinDr0p can be used to determine if information exchange through a website and a phone call are truly independent. Finally, PinDr0p could also be used by law enforcement agencies for call forensics.

5.5 Conclusion

Caller-ID has long been viewed as a reliable means of identifying the source of a call. However, this mechanism is now easily spoofable through a variety of free and low-cost techniques. In this paper, we take a first step towards a mechanism capable of determining call provenance — the source and the path taken by a call. We leverage attributes of the audio delivered to the receiver, including characteristics of the applied codes, packet loss profiles and bit error rates. We use these measurable elements to identify the codecs applied to incoming calls passing through as many as three intermediary types of telephony networks with a 91.6% accuracy. Moreover, fingerprints for specific sources were identified with between 90% and 100% accuracy with one and five training sets, respectively. This demonstrates that PinDr0p makes VoIP-based phishing attacks harder and provides an important first step towards a Caller-ID alternative. In our quest to create effective identities, we extend this work in the next chapter to obtain geographical information about a call.

CHAPTER VI

LONDON CALLING: EXTENDING CALL PROVENANCE TO DETECT GEOGRAPHY OF A CALLER

From Chapter 3, we realize that there are situations where we need more information than just identity. In the absence of common signaling and the impracticality of introducing elements within the telecommunication core, obtaining social network linkages for calls that traverse through multiple networks is infeasible. From the previous chapter we are encouraged by audio artifacts revealing the type of network a call traverses through. We now look at extending call provenance to determine the geography of a caller. Geography is attractive as it provides both organizations and consumers vital information about the legitimacy of a call. Financial institutions have expressed to us that geography can be used in conjunction with information that they record to determine if a particular call is fraudulent. For example, knowing that a customer call is coming from eastern Europe when a credit card transaction by the same customer was recorded a couple of hours back at Atlanta, provides a strong indication that the call is fraudulent. Towards determining geography, we need to first identify artifacts that are specific to certain paths. We then use these artifacts to see if we can group all calls coming over those paths. To understand why this is possible we first look at the notion of timbre of sound.

6.1 Timbre of Call Path

Timbre refers to the texture that is introduced into sound as a result of being produced by a specific sound production unit. For example, people with a keen ear are able to distinguish the same note at the same pitch and the same loudness produced by a

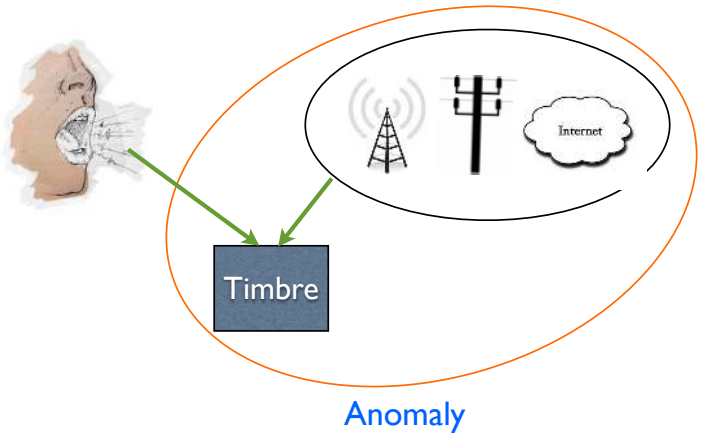


Figure 28: Anomalies in timbre are due to the call path. We hypothesize this will provide an indication of the path that a call takes.

Fender Stratocaster versus a Les Paul guitar. When a call goes over a certain path we hypothesize that the path adds a certain timbre to the call that can be used to identify geography. However, since a call is predominantly human voice, the voice itself adds timbre to the call. Therefore timbre that cannot be added by a human voice is essentially timbre created by the call path. We therefore look for anomalies in the sound that could not have been created by a human voice and use that to profile the call path as shown in Figure 28.

6.2 Identifying Anomalies in Timbre

Timbre in sound depends to a large extent on the envelope or how sound varies over time and to some extent on the spectrum. Therefore, determining anomalies in timbre is reduced down to identifying anomalies in the spectral envelope of the sound. We use two techniques traditionally used in sound to capture the spectral envelope, linear predictive coding and the cepstrum and we discuss these techniques in the next sections.

6.2.1 LPC and Cepstrum

We used LPC in chapter 6 to detect concealed packet losses in the iLBC codec. LPC can be used to model the geometry of the vocal tract through the duration of the call. In particular, we use it to model the back of the throat, the middle of the throat and the mouth. Since these are controlled by human muscles they can (1) only take on certain shapes, e.g. the mouth can only be stretched to a certain extent, and (2) the geometry can only have a certain rate of change, e.g. the throat cannot be wide at one instant and constricted in the next instant. We look for anomalies in the geometry, namely, excessively large sizes for the tracts or excessively fast changes in the geometry. Since these can not be caused by human voice they are artifacts introduced by the path. Similar to the LPC, another technique used to detect excessively fast variations is the cepstrum which produces information about the rate of change of different spectrum bands. In essence we use both LPC and cepstrum to profile call paths.

6.2.2 Identifying Anomalies in Vocal Tract Using LPC

LPC is based on the source filter model of speech production, where the larynx (source) produces sound energy, which when voiced consists of a fundamental frequency (pitch) and its harmonics. This sound energy is then shaped (synthesis filters) by the vocal tract (throat and mouth) into enhanced frequency bands known as formants, which provide speech its intonation. For unvoiced speech, there is no fundamental frequency. Voiced speech is the sound produced when uttering a vowel while unvoiced speech corresponds to consonants. Therefore, LPC considers voiced and unvoiced segments separately when analyzing and synthesizing speech. The basic model of operation for LPC is that it considers a block of speech, decides whether it is voiced or unvoiced and then decides the pith and the synthesis filters as parameters. For voiced segment, to determine the pitch we use the average magnitude difference

function (AMDF) as proposed in government standard 1014, also known as LPC-10. For determining the filter coefficients, the LPC estimates the current sample $s^e(t)$ at time t by p previous samples as $s^e(t) = \sum_{i=1}^{p} a_i s(t-i)$, where a_i is the filter coefficient. These coefficients are chosen to minimize the difference between the estimated value of the sample and the true value of the sample. These coefficients then give an accurate representation of the shape of the vocal tract.

The quantization of filter coefficients creates a major problem since errors in the filter coefficients can lead to instability in the final vocal tract filter and an inaccurate output signal. Instead, we use the Levinson-Durbin algorithm to generate reflection coefficients that can be used to rebuild the filter coefficients. From the reflection coefficients, we can then derive the uniform cross-sectional areas of the different elements of the vocal tract. This allows us to model the back of the throat, the middle of the throat and the mouth for each segment of speech. Based on well defined models of speech production there are threshold values for how large these areas can be and what are possible configurations. We use these values to determine if there are violations of these thresholds or infeasible configurations. We also look at vocal tract variations from one block of speech to the next and see if there are any unnaturally quick variations. Both of these are essentially the anomalies in the the vocal tract that must have been created by the path and the number and nature of such anomalies becomes a profile for the path that is producing them.

6.3 Evaluation

To evaluate PinDr0p we had asked family and friends, of the coauthors of that paper, located all over the world to make a large number of calls to our testbed in Atlanta. This approach was cumbersome and labor intensive. To avoid this, we instead decided to capture path profiles by making calls (instead of receiving) to different phone numbers across the world. The audio stream coming from the call recipient to us (the

caller) is still traveling the call path and will still contain all the artifacts introduced by this path. We can then measure how well anomalies detected by the LPC and the cepstrum can be used to profile certain geographies. We therefore made calls from our phone testbed in Atlanta, Georgia to 10 countries which include Philippines, Mexico, Japan, Canada, Venezuala, Russia, South Africa, Australia and Great Britain.

To ensure that there were no humans being bothered by these calls, we made calls to customer service numbers with IVR systems. The customer service numbers were scraped from a variety of Internet sources which include customerservicenumbers.com, the yellow pages, government, banking and transportation organizations of each of these countries. We looked at the length of the message and used that as an approximation as to whether we reached an IVR or a human. In our experience, if the call reached a human, they would hang up the phone immediately while an IVR would mindlessly continue providing a set of options. Though we can use more rigorous techniques to determine if we reached an IVR or a human (ask the recipient to press a DTMF tone if they have received the call in error), we found this technique to be quick and effective. We identified 9 customer service numbers in each country and made 10 calls to each of these numbers resulting in 10 phone calls/number * 9 numbers/country * 9 countries = 810 calls. Out of these, only 751 calls were actually placed as certain calls were not completed (e.g. the line being busy). To ensure that these calls were actually going to the right country, we did not call any numbers that were toll free as these numbers are typically forwarded to some other number potentially not in the same country. All numbers scraped had area codes specific to that country. In each call we recorded 30 seconds of IVR audio and ran algorithms to detect anomalies in the LPC and cepstrum and then determined the extent and distribution of these anomalies. We then labeled all calls with the country that was being called and used a 10 fold cross validated neural network classifier to see if we can identify a country from the call provenance profile.

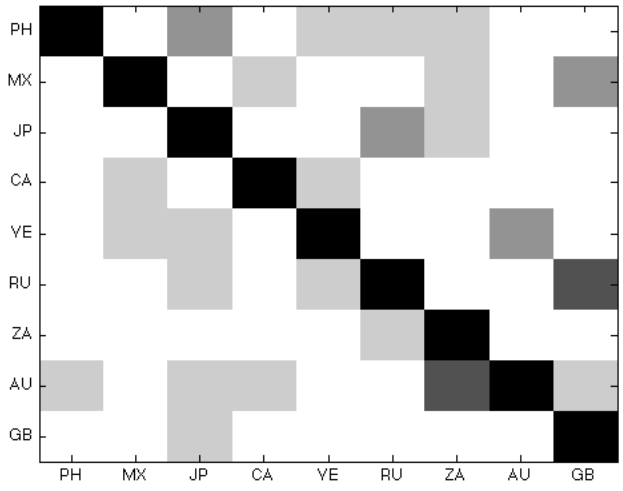


Figure 29: Confusion matrix for geography detection. Each country is represented by its two letter country code. Canada has the highest true positive rate while Australia has the lowest

6.4 Timbre of Call Path

The confusion matrix is as shown in Figure 29. 630 of the 751 calls are correctly classified providing an accuracy of 83.9%. As seen, certain countries are classified more accurately than others. For example, Canada has the highest true positive rate of 91% while Australia has the lowest true positive rate of 70.8%. Even more interesting are the misclassifications. We find Philippines is most often misclassified as Japan, while Russia is most often misclassified as Great Britain and Australia is misclassified as both Japan and South Africa. To understand these misclassifications, we further investigated the paths taken by these calls and looked at the undersea telecommunication cables between the US and other countries as shown in Figure 30. We find that 3 out of the 4 undersea cables from Phillipines pass through Japan in order to come to the US. Our calls to Russia were predominantly to West Russia whose telecommunication access to the US is through Great Britain, again explaining these misclassifications. Since we called both west and east Australia, these two regions take completely different routes to the US with cables going through Japan,

Figure 30: Undersea cables between the US and other countries. Though not shown, the cables travel across either the Atlantic or the Pacific to reach the US.

Great Britain and South Africa. However the undersea cables do not explain some of the other misclassifications, like Mexico being misclassified as Great Britain. As part of future work, we are investigating these to see what other effects are in place.

We further tried grouping countries into continents: Phillipines and Japan into Asia, Canada, Venezuala and Mexico into America, Great Britain and Russia into Europe and South Africa as Africa and used a neural network to predict the continent using 10 fold cross validation. We find we are able to classify 618 of the 679 calls correctly (we did not consider Australia) giving us an accuracy of 91%. This is because many of the misclassifications that arose due to common undersea cable from continents can be avoided. As part of future work we would like to see if we can use clustering algorithms to automatically determine a tradeoff between size of region and desired accuracy.

6.5 Conclusion

There are artifacts in call audio that can help us estimate the geography of a call. We used LPC and cepstrum to detect anomalies in timbre due to the call paths and

achieved an accuracy of over 83%. This provides additional information to determine if a call will result in a desirable interaction or not.

CHAPTER VII

CONCLUSION AND FUTURE WORK

The thesis hypothesized that it was possible to create effective identities in a converged telecommunication landscape that reduced both VoIP spam and Caller ID spoofing. We also wanted to determine that additional information shared to create these effective identities were privacy preserving. Towards the creation of effective identities to combat threats that undermine trust in telephony, we explored multiple techniques to determine if a call would lead to a desirable interaction. This thesis makes the following contributions:

- **CallRank:** We proposed CallRank, a system that uses call duration in combination with social network linkages and a global reputation to determine if a user is a spammer or not. In particular, we introduced call duration based credentials as the uniform underlying mechanism to determine if a caller is a spammer. We then explored the use of SNs based on the call credentials to allow two users to make a call. When SN linkages are unavailable between users, we used a variation of the Eigentrust algorithm to assign global reputations based on call durations. We finally performed a detailed evaluation of CallRank and show that we are able to achieve low false negative and low false positive rates even in the presence of a significant fraction of spammers.

- **Privacy Preserving Grapevines:** Though CallRank was effective in dealing with VoIP spam it shared sensitive information to prove the existence of a social network path. To address this limitation, we created Privacy Preserving Grapevines which proves the existence of a social network path without revealing the participants of that path. Specifically, we identified the requirements

for a framework that allows a new user, Bob, to prove the existence of SN call path between him and Alice, without revealing the actual path. In addition, the framework allows us to capture Alice's willingness to continue communicating with Bob. We then created a transferable single use token mechanism that extends delegatable anonymous credentials[36] with techniques from E-Cash[47] to realize this framework. We implemented this framework using the PBC library and experimentally evaluated the costs associated with its operations. Finally, we applied this framework to a VoIP setting and demonstrated that it can combat the spam problem with low false positive and false negative rates. CallRank and the privacy preserving extension to it show that within VoIP systems we can create effective identities that are able to reduce VoIP spam while ensuring the privacy of the users of the system.

- **Call Provenance:** We then broadened our focus and considered the entire telecommunication landscape. Here we observe that Caller-ID has long been viewed as a reliable means of identifying the source of a call. However, this mechanism is now easily spoofable through a variety of free and low-cost techniques. To address this we created mechanisms capable of determining call provenance — the source and the path taken by a call. In particular, we showed that the received call audio provides extractable features that are strong identifiers of the networks that the call has traversed, allowing us to determine the provenance of a call. These include degradations (packet loss in VoIP) and noise characteristics of codecs unique to each network. We developed a multi-label machine learning classifier based on the extracted features to correctly identify the provenance of an incoming call with 91.6% accuracy with as little as 15 seconds of audio. Because PinDr0p does not rely on metadata available in some networks (e.g., VoIP) or cryptography, it is more readily deployable across the diverse devices and networks that make up modern telephony systems. We

made calls using PSTN phones, cellular phones, Vonage, Skype and other soft phones from locations across the world and are able to distinguish between them with 90% accuracy with only a small sample being labeled. As we increase the number of such labels we are able to distinguish between these calls with 100% accuracy. This demonstrates that PinDr0p makes VoIP-based phishing attacks harder and provides an important first step towards a Caller-ID alternative. In addition to identity, we then extend provenance to obtain additional information in the form of geography of a call. Though not as powerful as identifying the social network linkages, this information is useful for financial institutions to determine the likelihood of a call to be coming from one of their customers. For example, this geography information can raise a flag when a user who has recently bought an item with a credit card in Atlanta, GA and a couple of hours later calls from a location across the Atlantic. This shows how effective identities can be use in the broader context of preventing Caller ID spoofing and even detecting potentially fraudulent calls.

7.1 Future Work

Through this thesis we have created effective identities in both VoIP networks and across the broader telecommunication landscape. This has opened up a large number of future avenues that we can pursue. These include:

- **Improvements to systems created:** Within Privacy Preserving Grapevines we found that we could only accommodate users four hops away as beyond that call setup times become unreasonable. We need to explore information available from real social networks to determine how often do people get introduced to users who are more that four hops away and what is the likelihood of spammers breaking such systems. Within PinDr0p we can improve the robustness of the

system through a number of extensions. While admittedly difficult, an adversary capable of replicating all of the codecs and path characteristics associated with the path between a legitimate source and target receiver would potentially be able to be identified as the profiled source. This process not only implies that the adversary has correctly guessed all of the codecs applied by intermediary hops, but that they can ensure that their traffic exhibits similar packet loss, bit error and noise characteristics as a legitimate connection. *This is exceptionally difficult as an adversary, for example, can not decrease the packet loss characteristics of an intermediary network that they do not control.* Our approach therefore represents a significant improvement over the current state of the art.

While we currently detect the presence of as many as three different codecs applied to audio, our mechanisms do not uncover the order in which the codecs were applied. Determining codec order is an extremely difficult problem on the surface. Knowledge of this ordering will make spoofing attempts by an adversary located off the path more difficult.

Finally, we are interested in extending our analysis to include a larger number of intermediary networks. While highly uncommon, it is possible that some international calls may be transcoded by as many as five different codecs while in flight between their source and destination. The repeated decoding and encoding of audio information drastically reduces its quality at the receiver end of the call and may also obscure the presence of the intermediary networks given the elevated noise levels present in the sample.

- **Identifying other categories of contextual information to create effective identities:** Within VoIP systems we used social network linkages and global reputation to provide additional information and within the telecommunication networks we used network type and geography. We would like to

explore other categories of contextual information that can be used to determine whether a call is beneficial or potentially malicious. We know that there are a large number of instances where people meet at conferences or classrooms and exchange phone numbers. In such cases there are potentially no social network linkages and yet calls initiated after such a meeting are legitimate. Capturing such social interactions would be extremely useful. One way to do this is to extend the system to consider cross channel information. For example all conference attendees can be considered a group and potential email interactions that register a user into a conference can be used to detect group membership. Therefore, we can use prior information from email interactions, create credentials out of these interaction and use them to determine the legitimacy of a subsequent call. In addition, we would like to explore other contextual information that can be extracted purely from call audio such as the actual telecommunication device (e.g., Skype softphone versus IPhone) being used in a call.

- **Effective identities in social networks:** We believe that the benefits of effective identities are not restricted to telecommunication systems. Our notion of identifying a metric that truly captures interactions between users (e.g., call duration in CallRank) and using that to determine the legitimacy of a call can be applied to parallel systems. For example, within Twitter we have looked at using Twitter specific conversation constructs such as @-mentions and retweets to identify legitimate users from noise makers/spammers. We would like to further explore the use of effective identities in such systems.

REFERENCES

[1] "AOL Instant Messenger." http://dashboard.aim.com/aim. Last accessed Sep 18, 2009.

[2] "Caller ID FAQ v2.32 1st April 2004." `http://www.ainslie.org.uk/callerid/cli_faq.htm`. Last accessed Sep 18, 2009.

[3] "Comcast." http://www.comcast.com/. Last accessed Nov. 18, 2009.

[4] "Confirmed cases of SPIT." http://www.voipsa.org/pipermail/voipsec_voipsa.org/2006-March/001326.html. Last accessed Nov. 18, 2009.

[5] "Constant Guard New Update." `http://security.comcast.net/constantguard/`. Last accessed Sep 18, 2010.

[6] "Google talk." http://www.google.com/talk/. Last accessed Sep 18, 2009.

[7] "IDC Predicts Almost Half a Billion Worldwide Personal IP Communications Subscribers." http://www.reuters.com/article/pressRelease/idUS103578+06-May-2008+BW20080506. Last accessed Nov. 18, 2009.

[8] "PlanetLab." `http://www.planet-lab.org/`. Last accessed Sep 18, 2009.

[9] "Skype Caller Identification." http://www.skype.com/allfeatures/calleridentification/. Last accessed Sep 18, 2009.

[10] "Skype: Our anti-spam initiatives." http://share.skype.com/sites/en/2009/08/our_antispam_i
Last accessed Nov. 18, 2009.

[11] "Tencent qq." http://www.imqq.com/. Last accessed Sep 18, 2009.

[12] "The Definitive Resource on Caller ID Spoofing." http://www.calleridspoofing.info/. Last accessed Sep 18, 2009.

[13] "The Harvard Sentences." `http://www.cs.columbia.edu/~hgs/audio/harvard.html`. Last accessed Sep 18, 2009.

[14] "Vonage." http://www.vonage.com/. Last accessed Nov. 18, 2009.

[15] "Vonage slips to Comcast in VoIP subscribers." http://abcnews.go.com/Technology/story?id=3463194&page=1. Last accessed Nov. 18, 2009.

[16] "What's the Deal with Skype Spam?." http://www.voip-news.com/feature/Whats-the-deal-012808/. Last accessed Nov. 18, 2009.

[17] "Windows messenger." http://download.live.com/messenger. Last accessed Sep 18, 2009.

[18] "Yahoo! messenger." http://messenger.yahoo.com/. Last accessed Sep 18, 2009.

[19] "Yahoo to Support OpenID for its 248 Million Users, OpenID to Support Yahoo IDs." http://www.searchenginejournal.com/yahoo-to-support-openid-for-its-248-million-users-openid-to-support-yahoo-ids/6258/. Last accessed Sep. 18, 2009.

[20] "IEEE Recommended Pratice for Speech Quality Measurements," in *IEEE Transactions on Audio and Electroacoustics*, vol. 17, 1969.

[21] "The E-model, a Computational Model for Use in Transmission Planning," Tech. Rep. ITU-T G. 107, ITU-T, February 2003.

[22] "The Speex Codec." `http://www.speex.org/`, 2003. Last accessed Sep 18, 2009.

[23] "Xbox LIVE." `http://www.xbox.com/en-US/LIVE/`, 2005. Last accessed Sep 18, 2009.

[24] "Pairing based cryptography library." http://crypto.stanford.edu/pbc/, 2006. Last accessed Sep 18, 2009.

[25] "Presentation on Q1 2009 Earning Report of Ebay Inc.k." `http://www.slideshare.net/earningreport/presentation-on-q1-2009-earning-report-of-ebay-inc`, 2009. Last accessed Sep 18, 2009.

[26] "IP Phone – Cisco." `http://www.cisco.com/en/US/products/hw/phones/ps379/index.html`, 2010. Last accessed Sep 18, 2009.

[27] "Skype." `http://www.skype.com/`, 2010. Last accessed Sep 18, 2009.

[28] Anagnostakis, K. G. and Greenwald, M., "Exchange-based incentive mechanisms for peer-to-peer file sharing," in *24th International Conference on Distributed Computing Systems (ICDCS)*, 2004.

[29] Balasubramaniyan, V., Ahamad, M., and Park, H., "Callrank: Combating spit using call duration, social networks and global reputation," in *CEAS 2007 - The Fourth Conference on Email and Anti-Spam, 2-3 August 2007, Mountain View, California, USA*, 2007.

[30] Balasubramaniyan, V., Lee, Y., and Ahamad, M., "Privacy preserving grapevines: Capturing social network interactions using delegatable anonymous credentials." `http://smartech.gatech.edu/handle/1853/31036`, 2009. Last accessed Sep 18, 2009.

[31] Balasubramaniyan, V., Lee, Y., and Ahamad, M., "Privacy preserving grapevines: Capturing social network interactions using delegatable anonymous credentials," Tech. Rep. GT-CS-09-12, Georgia Institute of Technology, 2009.

[32] Balasubramaniyan, V. A., Poonawalla, A., Ahamad, M., Hunter, M. T., and Traynor, P., "Pindr0p: using single-ended audio features to determine call provenance," in *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, (New York, NY, USA), pp. 109–120, ACM, 2010.

[33] Barabási, A.-L., *Linked: The New Science of Networks*. Perseus Books Group, May 2002.

[34] Barmouta, A. and Buyya, R., "GridBank: a Grid Accounting Services Architecture (GASA) for distributed systems sharing and integration," in *Parallel and Distributed Processing Symposium (IPDPS)*, pp. 8 pp.+, 2003.

[35] Barnes, J. A., "Graph theory and social networks: A technical comment on connectedness and connectivity," *Sociology*, vol. 3, no. 2, 1969.

[36] Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., and Shacham, H., "Randomizable proofs and delegatable anonymous credentials," in *Crypto*, vol. 5677 of *LNCS*, pp. 108–25, Springer-Verlag, Aug. 2009.

[37] Belenkiy, M., Chase, M., Erway, C. C., Jannotti, J., Küpçü, A., Lysyanskaya, A., and Rachlin, E., "Making p2p accountable without losing privacy," in *ACM workshop on Privacy in electronic society (WPES)*, (New York, NY, USA), pp. 31–40, ACM, 2007.

[38] Benjelloun, O., Das, A., Alon, S., and Widom, H. J., "Uldbs: Databases with Uncertainty and Lineage," in *In VLDB*, pp. 953–964, 2006.

[39] Boneh, D., Boyen, X., and Shacham, H., "Short group signatures," in *Crypto*, LNCS, Springer-Verlag, 2004.

[40] Bose, R. and Frew, J., "Lineage Retrieval for Scientific Data Processing: A Survey," *ACM Computing Surveys (CSUR)*, vol. 37, no. 1, pp. 1–28, 2005.

[41] Boyd, D. M., "Friendster and publicly articulated social networking," in *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, (New York, NY, USA), pp. 1279–1282, ACM Press, 2004.

[42] Boykin, P. and Roychowdhury, V., "Leveraging social networks to fight spam," *IEEE Computer*, vol. 38, no. 4, pp. 61–68, 2005.

[43] Brian Krebs, "FCC May Confront ISPs on Bot, Malware Scourge." `http://krebsonsecurity.com/2010/10/fcc-may-confront-isps-on-bot-malware-scourge/`, 2010. Last accessed Sep 18, 2010.

[44] Broom, S. R., "VoIP Quality Assessment: Taking Account of the Edge-Device," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 14, no. 6, pp. 1977–1983, 2006.

[45] Buneman, P., Khanna, S., and Tan, W. C., "Why and Where: A Characterization of Data Provenance," in *Proceedings of the International Conference on Database Theory (ICDT)*, 2001.

[46] Bussard, L., Roudier, Y., and Molva, R., "Untraceable secret credentials: Trust establishment with privacy," in *PERCOMW '04: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, (Washington, DC, USA), p. 122, IEEE Computer Society, 2004.

[47] Camenisch, J., Hohenberger, S., and Lysyanskaya, A., "Compact e-cash," in *EUROCRYPT, volume 3494 of LNCS*, pp. 302–321, Springer-Verlag, 2005.

[48] Canard, S. and Gouget, A., "Anonymity in transferable e-cash," in *Applied Cryptography and Network Security, 6th International Conference, ACNS 2008, New York, NY, USA, June 3-6, 2008. Proceedings*, pp. 207–223, 2008.

[49] Canard, S., Gouget, A., and Traoré, J., "Improvement of efficiency in (unconditional) anonymous transferable e-cash," pp. 202–214, 2008.

[50] Carminati, B. and Ferrari, E., "Privacy-aware collaborative access control in web-based social networks," in *Proceeedings of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security*, (Berlin, Heidelberg), pp. 81–96, Springer-Verlag, 2008.

[51] Chaum, D., Fiat, A., and Naor, M., "Untraceable electronic cash," in *CRYPTO '88: Proceedings on Advances in cryptology*, (New York, NY, USA), pp. 319–327, Springer-Verlag New York, Inc., 1990.

[52] Chaum, D. and Pedersen, T. P., "Transferred cash grows in size," in *EUROCRYPT*, pp. 390–407, 1992.

[53] Chaum, D. and van Heyst, E., "Group signatures.," in *EUROCRYPT*, pp. 257–265, 1991.

[54] Chirita, P.-A., Diederich, J., and Nejdl, W., "Mailrank: using ranking for spam detection.," in *CIKM* (Herzog, O., Schek, H.-J., Fuhr, N., Chowdhury, A., and Teiken, W., eds.), pp. 373–380, ACM, 2005.

[55] Cox, L. P. and Noble, B. D., "Samsara: honor among thieves in peer-to-peer storage," *SIGOPS Operating Systems Review*, vol. 37, pp. 120–132, December 2003.

[56] Cui, Y. and Widom, J., "Practical Lineage Tracing in Data Warehouses," in *Proceedings of the 16th International Conference on Data Engineering (ICDE)*, (Washington, DC, USA), 2000.

[57] Dantu, R. and Kolan, P., "Detecting Spam in VoIP Networks," in *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop*, (Cambridge, MA), pp. 31–37, July 2005.

[58] Dantu, R. and Kolan, P., "Detecting spam in voip networks," in *SRUTI'05: Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop*, (Berkeley, CA, USA), pp. 5–5, USENIX Association, 2005.

[59] Davidson, S. B. and Freire, J., "Provenance and Scientific Workflows: Challenges and Opportunities," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2008.

[60] Ding, L., Kolari, P., Finin, T., Joshi, A., Peng, Y., and Yesha, Y., "On Homeland Security and the Semantic Web: A Provenance and Trust Aware Inference Framework," in *Proceedings of the AAAI SPring Symposium on AI Technologies for Homeland Security*, March 2005.

[61] Ding, L., Lin, Z., Radwan, A., El-Hennawey, M. S., and Goubran, R. A., "Non-Intrusive Single-Ended Speech Quality Assessment in VoIP," *Speech Commun.*, vol. 49, no. 6, pp. 477–489, 2007.

[62] Dodis, Y. and Yampolskiy, A., "A verifiable random function with short proofs and keys," in *Proceedings of the Workshop on Theory and Practice in Public Key Cryptography*, 2005.

[63] Donato, D., Paniccia, M., Selis, M., Castillo, C., Cortese, G., and Leonardi, S., "New metrics for reputation management in p2p networks," in *AIRWeb '07: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, (New York, NY, USA), pp. 65–72, ACM, 2007.

[64] Douceur, J. R., "The sybil attack," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, (London, UK), pp. 251–260, Springer-Verlag, 2002.

[65] Douceur, J. R., "The sybil attack," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, (London, UK), pp. 251–260, Springer-Verlag, 2002.

[66] European Broadcasting Union, "Audio Contribution over IP." `http://www.ebu-acip.org/`. Last accessed Sep 18, 2009.

[67] Federal Communications Commission, "Rules and Regulations Implementing the Telephone Consumer Protection Act of 1991," tech. rep., 2003. Last accessed Sep 18, 2009.

[68] Federal Trade Commision, "Consumer Sentinel Network Data Book, January - December 2009." http://www.ftc.gov/sentinel/reports/sentinel-annual-reports/sentinel-cy2009.pdf, 2010. Last accessed Sep 18, 2009.

[69] Feldman, M., Papadimitriou, C., Chuang, J., and Stoica, I., "Free-riding and whitewashing in peer-to-peer systems," *Selected Areas in Communications, IEEE Journal on*, vol. 24, pp. 1010–1019, May 2006.

[70] Foster, I. T., Vöckler, J.-S., Wilde, M., and Zhao, Y., "Chimera: AVirtual Data System for Representing, Querying, and Automating Data Derivation," in *Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM)*, 2002.

[71] Franklin, J., McCoy, D., Tabriz, P., Neagoe, V., Randwyk, J. V., and Sicker, D., "Passive Data Link Layer 802.11 Wireless Device Driver Fingerprinting," in *Proceedings of the USENIX Security Symposium (SECURITY)*, 2006.

[72] FreeCallerIDSpoofing.com, "Free Caller ID Spoofing - Caller ID Changer - How to Make any Number You Want Show up on a Caller ID." http://freecalleridspoofing.com/, 2010. Last accessed Sep 18, 2009.

[73] Fuchsbauer, G., Pointcheval, D., and Vergnaud, D., "Transferable anonymous constant-size fair e-cash." Cryptology ePrint Archive, Report 2009/146, 2009. Last accessed Sep 18, 2009.

[74] Ghanbari, M., *Standard Codecs: Image Compression to Advanced Video Coding*. The Institution of Engineering and Technology, 2003.

[75] Global IP Solutions, "The Internet Low Bitrate Codec (ILBC)." http://tools.ietf.org/html/rfc3951, 2004. Last accessed Sep 18, 2009.

[76] Golub, G. H. and Van Loan, C. F., *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)*. The Johns Hopkins University Press, October 1996.

[77] Granovetter, M., "The strength of weak ties: A network theory revisited," *Sociological Theory*, vol. 1, pp. 201–233, 1983.

[78] Groth, J. and Sahai, A., "Efficient non-interactive proof systems for bilinear groups," in *Advances in Cryptology EUROCRYPT 2008*, pp. 415–432, 2008.

[79] Groth, P., Moreau, L., and Luck, M., "Formalising a Protocol for Recording Provenance in Grids," in *Proceedings of the UK OST e-Science Third All Hands Meeting 2004 (AHM'04)*, 2004.

[80] GSM, "GSM-FR: GSM Full Rate (GSM 06.10)." http://www.3gpp.org/FTP/Specs/html-info/0610.htm, 1995. Last accessed Sep 18, 2009.

[81] Hamadeh, I. and Kesidis, G., "A Taxonomy of Internet Traceback," *International Journal of Secururity and Networks*, vol. 1, no. 1/2, pp. 54–61, 2006.

[82] Hansen, M., Hansen, M., Mller, J., Rohwer, T., Tolkmit, C., and Waack, H., "Developing a legally compliant reachability management system as a countermeasure against spit," in *Proceedings of Third Annual VoIP Security Workshop*, (Berlin, Germany), Jun 2006.

[83] Hansen, T., Crocker, D., and Hallam-Baker, P., "Domainkeys identified mail (dkim) message signing service overview," Mar 2007. IETF-DRAFT draft-ietf-dkim-overview-04.txt.

[84] Hausheer, D., *PeerMart: Secure Decentralized Pricing and Accounting for Peer-to-Peer Systems*. PhD thesis, ETH Zurich, Aachen, Germany, Mar. 2006.

[85] Houmansadr, A., Kiyavash, N., and Borisov, N., "RAINBOW: A Robust And Invisible Non-Blind Watermark for Network Flows," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2009.

[86] Intel, "Intel Integrated Performance Primitives Library." `http://software.intel.com/en-us/intel-ipp/`. Last accessed Sep 18, 2009.

[87] Ioannidis, J., Ioannidis, S., Keromytis, A. D., and Prevelakis, V., "Fileteller: Paying and getting paid for file storage," in *Sixth International Conference on Financial Cryptography*, pp. 282–299, 2002.

[88] Jagadish, H. V. and Olken, F., "Database Management for Life Sciences Research," *ACM SIGMOD Record*, vol. 33, no. 2, pp. 15–20, 2004.

[89] Jennings, C., Peterson, J., and Watson, M., "Private extensions to the session initiation protocol (sip) for asserted identity within trusted networks," 2002.

[90] JK Audio - Telephone Audio Interface Products, "THAT-1: Telephone Handset Audio Tap." `http://www.jkaudio.com/that-1.htm`, 2009. Last accessed Sep 18, 2009.

[91] Joachims, T., "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," 1997.

[92] Justin Pritchard, "Credit Card Fraud - Credit Card Phone Activation Scam." `http://banking.about.com/od/securityandsafety/a/creditcardfraud.htm`. Last accessed Sep 18, 2009.

[93] Kamvar, S. D., Schlosser, M. T., and Garcia-Molina, H., "The eigentrust algorithm for reputation management in p2p networks," in *Proc. 12th International World Wide Web Conference*, (Budapest, Hungary), May 2003.

[94] Kawamoto, D., "SEC Filing Acknowledges 'Pretexting' in HP Board Probe." http://news.cnet.com/SEC-filing-acknowledges-pretexting-in-HP-board-probe/2100-1014_3-6112710.html, 2006. Last accessed Sep 18, 2009.

[95] Kekäläinen, J., "Binary and graded relevance in ir evaluations: comparison of the effects on ranking of ir systems," *Inf. Process. Manage.*, vol. 41, no. 5, pp. 1019–1033, 2005.

[96] Ken Paterson, "Credit Card Issuer Fraud Management." http://www.sas.com/news/analysts/mercator_fraud_1208.pdf, 2008. Last accessed Sep 18, 2009.

[97] Keynote Systems, Inc, "Internet Health Report." http://www.internetpulse.net/Main.aspx?Metric=PL, 2010. Last accessed Sep 18, 2009.

[98] Kim Tae-gyu, "Voice Phishing Getting Sophisticated." http://www.koreatimes.co.kr/www/news/biz/2010/01/123_43084.html, 2009. Last accessed Sep 18, 2009.

[99] Kiyavash, N., Houmansadr, A., and Borisov, N., "Multi-flow Attacks Against Network Flow Watermarking Schemes," in *Proceedings of the USENIX Security Symposium (SECURITY)*, 2008.

[100] Kong, L., Balasubramaniyan, V. A., and Ahamad, M., "A lightweight scheme for securely and reliably locating sip users," in *1st IEEE Workshop on VoIP Management and Security*, (Vancouver, Canada), Apr 2006.

[101] Kulbak, Y. and Bickson, D., *The eMule protocol specification*, January 2005.

[102] Lee, M. and McGowan, J. W., "Method and Apparatus for the Detection of Previous Packet Loss in Non-Packetized Speech." http://www.patentstorm.us/patents/7379864.html, May 2008. Last accessed Sep 18, 2009.

[103] Liebau, N., Darlagiannis, V., Mauthe, A., and Steinmetz, R., *Token-Based Accounting for P2P-Systems*. 2005.

[104] Linda McGlasson, "Vishing Scam: Four More States Struck." http://www.bankinfosecurity.com/articles.php?art_id=2138, 2010. Last accessed Sep 18, 2009.

[105] Lysyanskaya, A., Rivest, R. L., and Sahai, A., "Pseudonym systems," in *Proceedings of SAC 1999, volume 1758 of LNCS*, pp. 184–199, Springer Verlag, 1999.

[106] Malfait, L., Berger, J., and Kastner, M., "P.563 - The ITU-T Standard for Single-Ended Speech Quality Assessment," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 14, no. 6, pp. 1924–1934, 2006.

[107] MATHWORKS, "Simulink - Simulation and Model-Based Design." http://www.mathworks.com/products/simulink/. Last accessed Sep 18, 2009.

[108] MCCALLUM, A. K., "Multi-label text classification with a mixture model trained by em," in *AAAI 99 Workshop on Text Learning*, 1999.

[109] MILES, S., GROTH, P., BRANCO, M., and MOREAU, L., "The Requirements of Recording and Using Provenance in e-Science Experiments," *Journal of Grid Computing*, vol. 5, no. 1, 2007.

[110] MIYAJI, NAKABAYASHI, and TAKANO, "New explicit conditions of elliptic curve traces for fr-reduction," *TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems*, 2001.

[111] MOYER, T., BUTLER, K., SCHIFFMAN, J., MCDANIEL, P., and JAEGER, T., "Scalable Web Content Attestation," in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2009.

[112] PANG, J., GREENSTEIN, B., GUMMADI, R., SESHAN, S., and WETHERALL, D., "802.11 User Fingerprinting," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2006.

[113] PATANKAR, P., NAM, G., KESIDIS, G., and DAS, C. R., "Exploring anti-spam models in large scale voip systems," in *ICDCS '08: Proceedings of the 2008 The 28th International Conference on Distributed Computing Systems*, (Washington, DC, USA), pp. 85–92, IEEE Computer Society, 2008.

[114] PAXSON, V., "End-to-end routing behavior in the Internet," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 5, p. 56, 2006.

[115] PHONEGANGSTER.COM, "Anonymous Caller ID Spoofing Cards." http://www.phonegangster.com, 2010. Last accessed Sep 18, 2009.

[116] POUWELSE, J. A., GARBACKI, P., WANGAND, YANG, J., IOSUP, A., EPEMA, D., REINDERS, M., VAN STEEN, M. R., and SIPS, H. J., "Tribler: A social-based based peer to peer system," in *5th Int'l Workshop on Peer-to-Peer Systems (IPTPS)*, Feb 2006.

[117] PRIJONO, B., "PJSIP." http://www.pjsip.org/. Last accessed Sep 18, 2009.

[118] RAMACHANDRAN, A., BHANDANKAR, K., TARIQ, M. B., and FEAMSTER, N., "Packets with Provenance." http://www.cc.gatech.edu/research/reports/GT-CS-08-02.pdf, May 2008. Last accessed Sep 18, 2009.

[119] RAMAKRISHNAN, N., KELLER, B. J., MIRZA, B. J., GRAMA, A. Y., and KARYPIS, G., "Privacy risks in recommender systems," *IEEE Internet Computing*, vol. 5, pp. 54–62, November 2001.

[120] Rebahi, Y. and Sisalem, D., "Sip service providers and the spam problem," in *2nd Workshop on Securing Voice over IP*, (Washington DC, USA), Jun 2005.

[121] Research conducted by Javelin Strategy and Research, "2009 LexisNexis True Cost of Fraud Study." `http://www.riskfinance.com/RFL/Merchant_Card_Fraud_files/LexisNexisTotalCostFraud_09.pdf`, 2009. Last accessed Sep 18, 2009.

[122] Research conducted by Javelin Strategy and Research, "Identity Fraud Continues to Rise New Accounts Fraud Drives Increase; Consumer Costs at an All-Time Low." `https://www.javelinstrategy.com/research/brochures/Brochure-170`, 2010. Last accessed Sep 18, 2009.

[123] Resnick, P. and Zeckhauser, R., "Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system," in *The Economics of the Internet and E-Commerce* (Baye, M. R., ed.), vol. 11 of *Advances in Applied Microeconomics*, Elsevier Science, 2002.

[124] Rivest, R. L., Shamir, A., and Tauman, Y., "How to leak a secret.," in *ASIACRYPT* (Boyd, C., ed.), vol. 2248 of *Lecture Notes in Computer Science*, pp. 552–565, Springer, 2001.

[125] Rix, A. W., Beerends, J. G., Hollier, M. P., and Hekstra, A. P., "Perceptual Evaluation of Speech Quality (PESQ)-A New Method for Speech Quality Assessment of Telephone Networks and Codecs," in *Proceedings of the IEEE Internation Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Washington, DC, USA), 2001.

[126] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E., "Sip: Session initiation protocol," Jun 2002. RFC 3261.

[127] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E., "SIP: Session Initiation Protocol." RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393.

[128] Rosenberg, J. and Jennings, C., "The session initiation protocol (sip) and spam," Feb 2007. IETF-DRAFT draft-ietf-sipping-spam-04.txt.

[129] Ruggero Contu and Earl Perkins, "Market Trends: Identity and Access Management Market, Worldwide 2007-2013." `http://www.gartner.com/DisplayDocument?ref=clientFriendlyUrl&id=1297628`, 2010. Last accessed Sep 18, 2009.

[130] Santis, A. D. and Yung, M., "Crptograpic applications of the non-interactive metaproof and many-prover systems," in *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, pp. 366–377, 1990.

[131] Savage, S., Wetherall, D., Karlin, A., and Anderson, T., "Practical Network Support for IP Traceback," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 295–306, 2000.

[132] Shaw, R., "Four reasons why vonage ipos email and phone pitch is the wrong strategy," *ZDNet*, 2006.

[133] Sherr, M., Cronin, E., Clark, S., and Blaze, M., "Signaling Vulnerabilities in Wiretapping Systems," *IEEE Security & Privacy Magazine*, vol. 3, pp. 13–25, November 2005.

[134] Shin, D. and Shim, C., "Voice spam control with gray leveling," in *2nd Workshop on Securing Voice over IP*, (Washington DC, USA), Jun 2005.

[135] Shin, D., Ahn, J., and Shim, C., "Progressive multi gray-leveling: a voice spam protection algorithm," *IEEE Network*, vol. 20, no. 5, pp. 18–24, 2006.

[136] Shue, C. A., Gupta, M., Lubia, J. J., Kong, C. H., , and Yuksel, A., "Spamology: A study of spam origins," in *Conference on Email and Anti Spam (CEAS)*, 2009.

[137] Simmhan, Y. L., Plale, B., and Gannon, D., "A Survey of Data Provenance in E-Science," *ACM SIGMOD Record*, vol. 34, no. 3, pp. 31–36, 2005.

[138] Sirivianos, M., Park, J. H., Yang, X., and Jarecki, S., "Dandelion: cooperative content distribution with robust incentives," in *ATC'07: 2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference*, (Berkeley, CA, USA), pp. 1–14, USENIX Association, 2007.

[139] Slay, J. and Simon, M., "Voice over IP Forensics," in *Proceedings of the International Conference on Forensic Applications and Techniques in Telecommunications, Information, and Multimedia (e-Forensics)*, 2008.

[140] Snoeren, A. C., Partridge, C., Sanchez, L. A., Jones, C. E., Tchakountio, F., Schwartz, B., Kent, S. T., and Strayer, W. T., "Single-Packet IP Traceback," *IEEE/ACM Transactions on Networking (TON)*, vol. 10, pp. 721–734, December 2002.

[141] SpoofApp.com, "SpoofApp: Caller ID Spoofing For Your Mobile Phone." `http://www.spoofapp.com/`, 2010. Last accessed Sep 18, 2009.

[142] Srivatsa, M., Liu, L., and Iyengar, A., "Preserving Caller Anonymity in Voice-over-IP Networks," in *Proceedings of the IEEE Symposium on Security and Privacy (OAKLAND)*, (Washington, DC, USA), 2008.

[143] State of Kansas Supreme Court, "State v. Schuette." 273 Kansas 59, 44 P.3d 459), 2002. Last accessed Sep 18, 2009.

[144] Stenehjem, W., "Too Good To Be True: A Column on Consumer Trust Issues by Attorney General Wayne Stenehjem's Consumer Protection and Antitrust Division." www.ag.state.nd.us/tgtbt/2008/03-05-08.pdf, 2008. Last accessed Sep 18, 2009.

[145] Sterman, B., "A security model for spit prevention," in *2nd Workshop on Securing Voice over IP*, (Washington DC, USA), Jun 2005.

[146] Tae, H., Kim, H. L., Seo, Y. M., Choe, G., Min, S. L., and Kim, C. S., "Caller Identification System in the Internet Environment," in *In Proceedings of the USENIX Security Symposium (SECURITY)*, 1993.

[147] Takahashi, A., Kurashima, A., and Yoshino, H., "Objective Assessment Methodology for Estimating Conversational Quality in VoIP," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 14, no. 6, pp. 1984–1993, 2006.

[148] Tempich, C., Staab, S., and Wranik, A., "Remindin': semantic query routing in peer-to-peer networks based on social metaphors," in *WWW '04: Proceedings of the 13th international conference on World Wide Web*, (New York, NY, USA), pp. 640–649, ACM, 2004.

[149] The International Telecommunication Union, "G.711: Pulse Code Modulation (PCM) of Voice Frequencies." http://www.itu.int/rec/T-REC-G.711/e, 1972. Last accessed Sep 18, 2009.

[150] The International Telecommunication Union, "G.729: Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear Prediction." http://www.itu.int/rec/T-REC-G.729/e, 1996. Last accessed Sep 18, 2009.

[151] The International Telecommunication Union, "G.711 Appendix I." http://www.itu.int/rec/T-REC-G.711/recommendation.asp?lang=en&parent=T-REC-G.711-199909-I!AppI, 1999. Last accessed Sep 18, 2009.

[152] The International Telecommunication Union, "Recommendation P.563 - Single Ended Method for Objective Speech Quality Assessment in Narrow-Band Telephony Applications." http://www.itu.int/itudoc/itu-t/aap/sg12aap/history/p563/index.html, 2004. Last accessed Sep 18, 2009.

[153] The Machine Learning and Knowledge Discovery Group at Aristotle University of Thessaloniki, "Mulan: An Open Source Library for Multi-Label Learning." http://mlkd.csd.auth.gr/multilabel.html, 2010. Last accessed Sep 18, 2009.

[154] Thigpen, W., Hacker, T. J., Mcginnis, L. F., and Athey, B. D., "Distributed accounting on the grid," in *In Proceedings of the 6th Joint Conference on Information Sciences*, pp. 1147–1150, 2002.

[155] Thom, G. A., "H.323: the multimedia communications standard for local area networks," *Communications Magazine, IEEE*, vol. 34, no. 12, pp. 52–56, 1996.

[156] Tschofenig, H., Peterson, J., Polk, J., Sicker, D., and Tegnander, M., "Using saml for sip," Jul 2005. IETF-DRAFT draft-tschofenig-sip-saml-04.txt.

[157] Tsoumakas, G. and Katakis, I., "Multi-label Classification: An Overview," *International Journal of Data Warehousing and Mining*, vol. 2007, pp. 1–13, 2007.

[158] Tsoumakas, G. and Vlahavas, I., "Random k-Labelsets: An Ensemble Method for Multilabel Classification," in *ECML '07: Proceedings of the 18th European conference on Machine Learning*, (Berlin, Heidelberg), pp. 406–417, Springer-Verlag, 2007.

[159] Turner, D. A., "A lightweight currency paradigm for the p2p resource market," in *7th International Conference on Electronic Commerce Research*, 2003.

[160] Union, I. T., "Network grade of service parameters and target values for circuit-switched services in the evolving isdn," 2004.

[161] Union, I. T., "Measuring the information society - the ict development index 2009," 2009.

[162] VISA Inc, "Fraud Alert - Personal Identification Number (PIN) Attacks." `http://usa.visa.com/download/merchants/20090205_pin_attacks.pdf`, 2009. Last accessed Sep 18, 2009.

[163] VoDaSec, "Spitting over the internet." http://www.vodasec.com/.

[164] VoIP Troubleshooter.com, "The Open Speech Repository." `http://www.voiptroubleshooter.com/open_speech/index.html`, 2010. Last accessed Sep 18, 2009.

[165] Wang, X., Chen, S., and Jajodia, S., "Tracking anonymous peer-to-peer VoIP calls on the Internet," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2005.

[166] Watts, D. J. and Strogatz, S. H., "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, April 1998.

[167] Wei, K., Smith, A. J., Chen, Y.-F. R., and Vo, B., "Whopay: A scalable and anonymous payment system for peer-to-peer environments," in *ICDCS '06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, (Washington, DC, USA), p. 13, IEEE Computer Society, 2006.

[168] Weisbaum, H., "Don't Get Hooked by Latest Phishing Scam." `http://www.msnbc.msn.com/id/18553590/`, 2007. Last accessed Sep 18, 2009.

[169] Woodruff, A. and Stonebraker, M., "Supporting Fine-grained Data Lineage in a Database Visualization Environment," in *Proceedings of the International Conference on Data Engineering (ICDE)*, (Washington, DC, USA), 1997.

[170] Wright, C., Ballard, L., Coull, S., Monrose, F., and Masson, G., "Spot Me if You Can: Recovering Spoken Phrases in Encrypted VoIP Conversations," in *Proceedings of the IEEE Symposium on Security and Privacy (OAKLAND)*, 2008.

[171] Wright, C., Ballard, L., Monrose, F., and Masson, G., "Language Identification of Encrypted VoIP Traffic: Alejandra y Roberto or Alice and Bob?," in *Proceedings of the USENIX Security Symposium (SECURITY)*, 2007.

[172] Xiang, Y., Zhou, W., Li, Z., and Zeng, Q., "On the Effectiveness of Flexible Deterministic Packet Marking for DDoS Defense," in *Network and Parallel Computing (NPC) Workshops*, 2007.

[173] Xie, Y., Sekar, V., Reiter, M., and Zhang, H., "Forensic Analysis for Epidemic Attacks in Federated Networks," in *Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols (ICNP)*, (Washington, DC, USA), pp. 43–53, 2006.

[174] Yang, B. and Garcia-Molina, H., "Ppay: micropayments for peer-to-peer systems," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, (New York, NY, USA), pp. 300–310, ACM, 2003.

[175] Yang, B., Sun, J. T., Wang, T., and Chen, Z., "Effective multi-label active learning for text classification," in *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, (New York, NY, USA), pp. 917–926, ACM, 2009.

[176] Zhang, Y. and Paxson, V., "Detecting Stepping Stones," in *Proceedings of the USENIX Security Symposium (SECURITY)*, 2000.

[177] Zhou, W., Cronin, E., and Loo, B. T., "Provenance-aware secure networks," in *Proceedings of the International Conference on Data Engineering Workshops (ICDE)*, 2008.