Socio-Technical Defense against Voice Spamming

PRAKASH KOLAN and RAM DANTU
University of North Texas, Denton, Texas

Voice over IP (VoIP) is a key enabling technology for migration of circuit-switched PSTN (Public Switched Telephone Network) architectures to packet-based networks. One problem of the present VoIP networks is filtering spam calls referred to as SPIT (Spam over Internet Telephony). Unlike spam in e-mail systems, VoIP spam calls have to be identified in real time. Many of the techniques devised for e-mail spam detection rely upon content analysis, and in the case of VoIP, it is too late to analyze the content (voice) as the user would have already attended the call. Therefore, the real challenge is to block a spam call before the telephone rings. In addition, we believe it is imperative that spam filters integrate human behavioral aspects to gauge the legitimacy of voice calls. We know that, when it comes to receiving or rejecting a voice call, people use the social meaning of trust, reputation, friendship of the calling party and their own mood. In this article, we describe a multi-stage, adaptive spam filter based on presence (location, mood, time), trust, and reputation to detect spam in voice calls. In particular, we describe a closed-loop feedback control between different stages to decide whether an incoming call is spam. We further propose formalism for voice-specific trust and reputation analysis. We base this formal model on a human intuitive behavior for detecting spam based on the called party's direct and indirect relationships with the calling party. No VoIP corpus is available for testing the detection mechanism. Therefore, for verifying the detection accuracy, we used a laboratory setup of several soft-phones, real IP phones and a commercial-grade proxy server that receives and processes incoming calls. We experimentally validated the proposed filtering mechanisms by simulating spam calls and measured the filter's accuracy by applying the trust and reputation formalism. We observed that, while the filter blocks a second spam call from a spammer calling from the same end IP host and domain, the filter needs only a maximum of three calls—even in the case when spammer moves to a new host and domain. Finally, we present a detailed sensitivity analysis for examining the influence of parameters such as spam volume and network size on the filter's accuracy.

This work was supported by the National Science Foundation under grants CNS-0627754 (Detecting Spam in IP Multimedia Communication Services), CNS-0516807 (Preventing Voice Spamming), CNS-0619871(Development of a Flexible Platform for Experimental Research in Secure IP Multimedia Communication Services), and CNS-0551694 (A Testbed for Research and Development of Secure IP Multimedia Communication Services).
Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
Authors' address: Department of Computer Science, University of North Texas, Denton, TX 76201; email: prk0002@cs.unt.edu; rdantu@unt.edu.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Applications*; H.4 [**Information System Applications**]: Communications Applications; K.4.1 [**Computers and Society**]: Public Policy Issues—*Abuse and crime involving computers*

General Terms: Human Factors, Security

Additional Key Words and Phrases: Trust, reputation, SPIT (Spam over IP Telephony), behavior, tolerance, SIP (Session Initiation Protocol)

ACM Reference Format:

Kolan, P. and Dantu, R. 2007. Socio-technical defense against voice spamming. ACM Trans. Autonom. Adapt. Syst. 2, 1, Article 2 (March 2007), 44 pages. DOI = 10.1145/1216895.1216897 http://doi.acm.org/10.1145/1216895.1216897

1. INTRODUCTION

Defending the country's telecommunication networks requires cooperation between service providers, equipment vendors, enterprises, and the government. Currently, VoIP infrastructure is being aggressively deployed in both enterprises and residential areas without due consideration of security aspects. Insecure deployment offers a clear recipe for a possible disaster to critical telecommunication infrastructures. However, little work appears in the literature on how to defend VoIP against attacks such as Denial of Service (DOS), session hijacking and termination, monitoring and eavesdropping, service disruption, toll fraud, identity fraud, and spamming. Also, the impact of vulnerabilities on a large-scale VoIP network (e.g., several millions of IP phones) is not well understood. Hence, it is imperative that we thoroughly investigate the vulnerabilities and threats to communities from deployment of real-time services like VoIP. All the threats need to be addressed before we deploy VoIP services on a mass scale because the lack of security could potentially disrupt the next generation voice communications.

The possibility of VoIP network replacing the PSTN network depends on enhancing the existing IP network to carry voice traffic. With the usage of IP network to carry voice traffic, existing problems on the IP network holds for the VoIP network too. One of the major issues that current IP networks face is controlling spam—the unsolicited (bulk) e-mail. Spam control has been perceived to be one of the most important problems of research with the traditional e-mail systems. A recent study ([Evett 2006]) indicates that over 40% of the e-mail circulating on the Internet nowadays is spam. Daily traffic is expected to rise above 68 billion messages per day, and more than half of it–63%–will be spam by 2007. The study estimates the spam costs for US corporations to reach $8.9 billion. With this quantity of spam messages circulating through the Internet each day, problems such as low availability and network congestion would not be a surprise. In VoIP networks, spam refers to unsolicited voice calls (usually referred to as SPIT—Spam over IP Telephony) that consume resources on both the end VoIP phones and intermediate VoIP infrastructure components. So far, we have no documented cases of SPIT because we have few people using VoIP services. But, with the rapid pace of deployment and the number of residential subscribers estimated to reach about 140 million users

in 2010 ([Rago 2006]), SPIT poses a major threat if the VoIP industry fails to adequately address prevention mechanisms. An analysis by Rosenberg et al. [2006] indicates that spammers find IP-based SPIT roughly three orders of magnitude cheaper to send than traditional circuit-based telemarketer calls. For example, consider a VoIP spammer generating random usernames, IP addresses or domain names to form a SIP (Session Initiation Protocol) URI (SIP identity) in the form *sip:username@ip_address* such as in *sip:alice@abcdef.com* (similar to e-mail addresses in e-mail domain). The spammer can then use this randomly generated SIP URI to generate SPIT calls (similar to the way an e-mail spammer generates random addresses for sending spam emails). While many techniques have been designed to avoid e-mail spam, such techniques often have limited application to avoid voice spam because of real time considerations. In addition, content filtering is not useful in VoIP spam analysis as media flows in after the two parties (i.e., calling party and the called party) have agreed upon to start the communication and would be too late to filter the call. Compare receiving an e-mail spam at 2:00 AM to receiving a junk voice call. The e-mail spam sits in our Inbox until we see it later in the morning. The junk voice call makes the phone to ring if it reaches the called party. This inability to filter VoIP calls poses a serious challenge of detecting spam in real time with the available signaling messages.

To realize our objective of receiving only genuine VoIP calls from any person anywhere in the world, we must replace static junk-call filtering mechanisms with adaptive learning systems. These systems, apart from learning spam behavior, should incorporate human behavioral models of how the call recipients (called parties) determine whether to answer call. For example, whenever a phone rings, depending on our state of mind, we determine whether the call is from a trusted party. If we do not know the calling party, we guess the calling party's reputation. After picking up the call, we ask the calling party some questions and move forward only when satisfied with the calling party's response. Similarly, our adaptive learning system uses an intelligent call admission control which takes into account the presence of the called party (e.g., state of mind, location, time), the rate of incoming calls from the calling party (by computing first and second order differentials), trust between calling party and the called party (estimated using Bayesian theory), and reputation graphs based on called party's social network. We have integrated these factors within our adaptive learning system to facilitate deciding whether to accept/reject a call or forward it to voice mail. We organize this article as follows: In Section 2, we present related work in trust and reputation computational models, e-mail spam filtering techniques, and SPIT detection methods. In Section 3, we propose a Voice Spam Detector (VSD) for analyzing incoming calls. In Section 4, we present an adaptive learning mechanism we integrated into the VSD for inferring spam behavior. Next, we present a trust and reputation formalism which takes into account human intuitive behavior in detecting spam based on direct (trust) and indirect (reputation) relationships with the calling party. In Section 5, we present experimental results when the formal model is integrated into VSD for spam analysis. In Section 6, we present a detailed sensitivity analysis on the accuracy of VSD with respect to parameters such as spam volume and network

size. Finally, we present a mechanism for increasing the filter‘s accuracy by integrating domain-wide knowledge in the spam analysis.

2. BACKGROUND

There exists lot of literature in the field of trust and reputation computation [Rahman and Hailes 1998; Lei and Shoja 2005; Marsh 1994; Cahill et al. 2003; Krukow and Neilson 2006; Zimmerman 1995; Wang and Vassileva 2003a, 2003b; Yu and Singh 2001, 2002; Mui et al. 2002; Zacharia et al. 1999; Sabater and Sierra 2005; Jøsang et al. 2006]. Rahman and Hailes [1998] presents a distributed trust model for computing trust of entities in online transactions. The trust model adopts a recommendation protocol where a source entity requests its trusted entities to give recommendations about a target entity for a given trust category. When the source entity receives all the recommendations, it computes a trust value to the target entity based on the received recommendations. Lei and Shoja [2005] present a distributed trust model organized as a Trust Delegation Tree (TDT) in e-commerce applications. This article presents a trust management model for computing different trust levels such as direct trust based on history, indirect trust based on trusted intermediaries' recommendations, and trust authorization levels using delegation certification chains. Marsh [1994] describes trust formalism for computing and updating trust between two agents. The trust mechanism involves the computation of types of trust such as basic trust (based on accumulated experiences), general trust independent of context, and situational trust that takes into account a specific situation. Cahill et al. [2003] presents a trust model for secure collaborations among pervasive entities. This article presents a trust model in which a source principal makes a decision whether to interact with a target principal based on current trust value he holds with the target principal for that particular action, and the risk in communicating with the target principal. Krukow and Nielsen [2006] discuss the importance of probabilistic models in logic reasoning. This article presents a probabilistic trust model for computing the predictive probability of a principal behavior that is, the probability with which the principal's next interaction will have a specific outcome. Zimmerman [1995] describes a web of trust model where users can exchange PGP keys among themselves to trust each other. The exchanged keys are signed by each user, and ant any time, the trustworthiness of users that have signed the keys can be established. However, the users are required to create their own security and trust policies. The web of trust model is not scalable and is not used with very large systems such as the Internet. Wang and Vassileva [2003a, 2003b] present a Bayesian trust model for inferring the trust of agents participating in an online transaction. The proposed trust mechanism involves deriving Bayesian Networks for trust inference and using the past transaction history to update the available trust information. In addition, the trust model incorporates a mechanism for evaluating the recommendations of other agents and updating its trust values towards them. Yu and Singh [2001, 2002] describe a trust framework for classifying agents to be trustworthy based on quality of recent transactions. The agents provide thresholds for differentiating trustworthiness of agents into

trustworthy, nontrustworthy and unclear classification groups. The framework assumes that an agent belongs to one of these groups when the probability of service between that group and latest group is greater than a give threshold. In addition, the framework gives preference to direct interaction information before taking into account indirect information from witnesses. Mui et al. [2002] present a computational model for inferring trust and reputation of a given source. The model infers trust as a dyadic quantity between trustor and trustee and is computed based on the reputation data of the trustee. This article further defines the reputation score as a quantity embedded in the social network of the trustor and past transactions with the trustee. Zacharia et al. [1999] and Zacharia and Maes [2000] present two reputation mechanisms "Sporas" and "Histos" for inferring the reputation of agents in an online community. Sporas is a reputation computation system where the agents rate each other after the completion of the transaction and the reputation values are updated. This update is less when the agent's reputation is high. Histos is a reputation computation mechanism that performs a recursive personalized rating inference of agents that have communicated with the target agent. This recursive inference is achieved by deriving a weighted graph with nodes as the agents and the links connecting them as personalized ratings given by the parent node to the child node of the link. Sabater and Sierra [2005] presents a survey on computational trust and reputation models that are of specific application in the field of distributed Artificial Intelligence. This article describes different classification aspects based on which the models can be classified and provides a review on sample models in the area of trust and reputation computation research. Jøsang et al. [2006] present a comprehensive survey of trust and reputation models in Internet transactions. The survey describes the trust and reputation semantics existing in the literature of trust and reputation inference. This article also describes the existing problems and solutions for aggregation trust and reputation metrics. All these trust and reputation inference techniques have been used to solve problems in different problem domains such as e-commerce, peer-to-peer networks, and spam filtering.

Spam filtering for the present day e-mail infrastructure has been well addressed in current literature [Sahami et al. 1998; Soonthornphisaj et al. 2002; Sakkis et al. 2003; Cohen 1996; Rigoutsos and Huynh 2004; Golbeck and Hendler 2004; Seigneur et al. 2004; Damiani et al. 2004; Wattson 2004; Foukia et al. 2006]. Designers of spam filters have used a wide variety of filtering mechanisms such as text classification, rule-based scoring systems, Bayesian filtering, pattern recognition, and identity verification. Sahami et al. [1998] describe a Bayesian trust model for filtering spam e-mails. This article proposes that incorporating domain specific features in addition to identifying various textual phrases, and probabilistically inferring the spam behavior of the constructed message vector leads to a more accurate spam analysis. Soonthornphisaj et al. [2002] presents a spam filtering technique that constructs the centroid vector of the incoming e-mail and checks for its similarity between the centroid vector of spam class and legitimate class e-mails. Sakkis et al. [2003] suggests probabilistic inference for calculating the mutual information index (MI) for feature selection. Using this, a vector of attributes having the highest MI scores is

constructed for spam identification. The memory-based algorithms then attempt to classify messages by finding similar previously received messages and using them for classification. Cohen [1996] recommends spam filtering based on a set of rules for identifying the message body content. Features of the message are identified and scored to compute the total spam score of the e-mail spam message and the messages having a score more than a given threshold is identified as a spam e-mail. Large quantities of spam and legitimate messages are used to determine the appropriate scores for each of the rules in the rule-based scoring systems. Rigoutsos and Huynh [2004] suggests pattern discovery scheme for identifying unsolicited e-mails by training the system with a large number of spam messages. The system matches the e-mail message with the available patterns; more the patterns are matched more is likelihood that the message is spam. Golbeck and Hendler [2004] presents an algorithm for inferring reputation relationships. This is achieved by constructing social network of people connected with each other. Every user in the social network is attributed with a reputation score. For a given message from the e-mail sender, the receiver infers the weighted average of its neighbor's reputation ratings to the e-mail sender. The neighbors in turn infer the weighted average of their neighbors' reputation rating for the e-mail sender and this continues until the e-mail sender is reached. Seigneur et al. [2004] discusses the establishment of identities of e-mail senders using Trustworthy e-mail addresses. The identities of the parties are established by exchanging hashes of previously exchanged e-mails. In turn, a C/R system is discussed for challenging the party to establish its identity. This article also discusses a Trust/Risk Security framework for inferring whether the incoming e-mail is spam or not using. The framework proposes to use a Bayesian spam filter for trust inference. In addition, the framework uses a static model for choosing a finite set of recommenders for making a recommendation. Identity verification mechanisms help in establishing that the caller is the person who he claims to be. However, identity-based mechanisms are not a complete solution for filtering spam especially in cases of dynamically changing behavior and preferences of people involved in communication. Damiani et al. [2004] suggests a P2P framework for detecting e-mail spam messages. A P2P network consisting of user-tier nodes, mailers (mail servers), and super peers exchange communication among themselves for tagging and updating incoming e-mail messages as spam. This is achieved by constructing message digests of incoming messages and checking for similarity with other spam message digests. Wattson [2004] presents a spam filtering mechanism based on sender identity verification and disposable e-mail addresses. The mechanism proposes a multistage architecture consisting of black- and white-listing procedures, sender identity verification, and challenge response systems. The cumulative inference of all the stages dictates whether the incoming e-mail is spam or not. Foukia et al. [2006] presents a collaborative framework for spam control in distributed administration domains. The framework involves mail servers collaborating with each other to exchange spam control information. The proposed framework involves spam control processes at both the incoming and outgoing mail servers. The servers that participate in this information exchange are rewarded and traffic restrictions are imposed on those e-mail servers

that do not participate. All the above proposed techniques attempt to classify incoming e-mail as spam based either on static rule-checking, learning spam behavior from the e-mail's contents, or through identity verification. However, these techniques cannot be directly used for filtering real-time junk voice. Unlike e-mail spam messages, VoIP calls are real-time and have to be filtered in real-time. In e-mail, people do not exhibit dynamic changes in behavior. Even if the people exhibit dynamic changes in behavior, the e-mails from them do not pose a nuisance to the end user. However, in case of VoIP, real-time voice calls from people exhibiting dynamic changes in behavior creates a lot of nuisance to the called party. This nuisance depends on the presence (mood or state of mind) depending on context (such as situational, temporal, and spatial) of the called party. The nuisance in this case is even more if the caller is unknown to the called party.

Little work exists on analyzing spam in VoIP networks. The SIP spam draft [Rosenberg et al. 2006] discusses spam in VoIP networks. Rebahi and Sisalem [2005] present a spam filtering solution using reputation relationships. The mechanism proposes to build a social network of people that can issue recommendations regarding a given source. Macintosh and Vinokurov [2005] present a statistical detection technique by analyzing the incoming traffic distribution of the VoIP calls. An abnormal deviation from the normal call distribution is considered to be SPIT traffic. Shin and Shim [2005] presents a spam filtering approach where the calls are analyzed based on their incoming rate. If the incoming rate is greater than predetermined short-term and long-term thresholds, the call is blocked and branded as spam. We believe that a spam solution for analyzing incoming voice calls is not only confined to limiting the rate of calls. The solution should also consider the social authenticity (trust and reputation), social connectivity, and the inherent need in accepting the incoming call based on called party's presence.

3. VOICE SPAM DETECTOR

The network diagram shown in Figure 1 is partitioned into network segments such as the call-receiving domain—VSD Domain (hereafter referred to as VSD_{domain}-circled part in Figure 1) that is, the domain for which the VSD acts as a spam detector (e.g., an enterprise network), and the call-generating domains (e.g., telemarketing companies) that is, the domains from which calls would be generated to an end user in the VSD_{domain} (e.g., an enterprise employee). This VSD_{domain} consists of all the VoIP users whose calls are analyzed by the VSD for spam behavior. VSD_{domain} can be scaled to different network sizes. For example, the VSD_{domain} can be an enterprise having a Class B network such as the domain www.unt.edu (University of North Texas with an IP address range 129.120.xxx.xxx), or a scaled down domain such as the computer science department at UNT (with an IP address range 129.120.60.xxx or 129.120.61.xxx). At any level, VSD analyzes and filters the call for the users inside the network (or domain). Calls are generated from an end user outside or inside VSD_{domain} through the VSD. Each IP phone in VSD_{domain} includes a SPAM button to allow the called party (callee) to give feedback to the VSD.

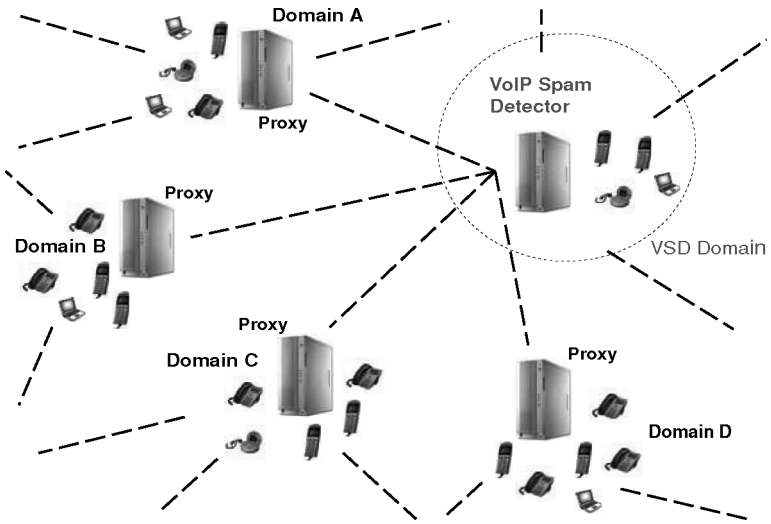

Fig. 1. Voice Spam Detector for computing the spam probability of incoming calls. The VSD can be deployed either in a VoIP proxy server (e.g., at the enterprise perimeter) or in an end VoIP phone. In any case, VSD analyzes the incoming and outgoing calls based on end users' preferences.

On receiving a call, VSD analyzes the *spam level* of the incoming call (the associated spam probability) using the VoIP spam-detection model presented in Section 4. The VSD then compares the call's computed spam level with a predetermined threshold value to decide whether to block or forward the call to the callee. The threshold value (permissible limit) is chosen by giving preference to legitimate calls over spam calls, that is, the number of spam calls that can be forwarded so as to minimize false positives (legitimate calls being blocked). The main aim of any spam-filtering technique should be to minimize false negatives (spam calls let in as legitimate) while keeping the false positives to zero.

The call processing depends on the callee's reaction to the incoming calls. The VSD_{domain} users are equipped with spam-recognition capabilities. The callee receives the call, experiences it, and gives feedback to the VSD about the nature of call (whether it is a spam call or a legitimate call). This feedback is as simple as pressing a spam button either during the call or just after termination. This feedback resembles the way e-mail users give feedback about spam e-mail by clicking a "SPAM" button on their web browser. The VSD learns about the spam behavior of call participants (such as user, host and domain) based on callee's feedback. If the callee responds with a feedback that the current call is spam, the VSD updates the calling party's (caller's) history for future spam analyses. Future calls from the caller will have a high spam probability and a higher chance of being stopped at VSD. On the other hand, if the callee responds with a positive experience (non-spam), the caller's history is updated to depict more legitimacy for next incoming calls from the caller.

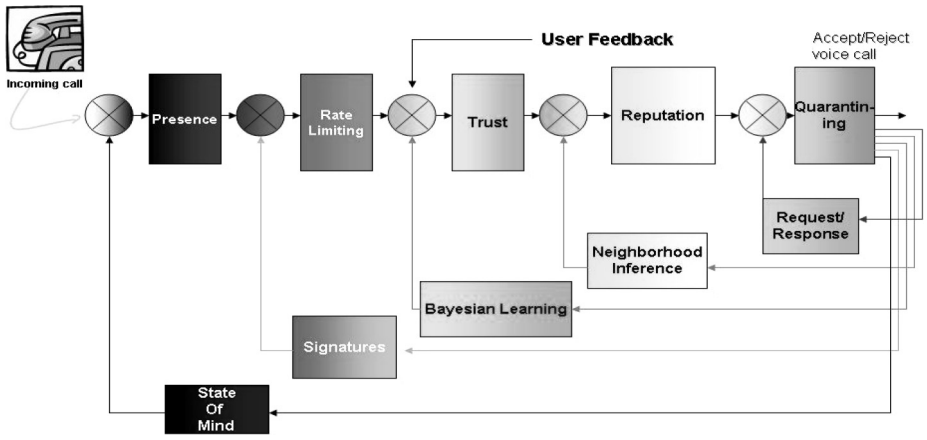

Fig. 2. Functional Elements of VSD. The diagram describes a logical view of a multi-loop feedback control process. Each stage employs a specific set of mechanisms for computing the incoming call's spam level. Each stage computes a spam level based on its specific features. A collective inference from all of the stages represents an incoming call's spam level.

4. VOIP SPAM DETECTION FRAMEWORK

VoIP spam detection cannot be achieved using a single detection method. The detection needs to occur at several stages of call processing to achieve a high degree of accuracy.

4.1 Architecture

The architecture for spam detection process should take into account the callee's preferences of wanted and unwanted calls, his presence of mind, the trust and reputation he has for the caller. The basic architecture for spam detection is shown in Figure 2. Each stage in the architecture diagram represents a technique based on which the call would be quarantined by employing a specific set of mechanisms and user feedback. Each stage computes the possibility that the call is spam and the collective inference of all stages provides a measure of the call's probability of being spam. Based on this inference, a decision is made to forward or quarantine the call.

4.2 Functional Elements in Voice Spam Detection

As depicted in Figure 2, a number of techniques are used to achieve a composite measure of a call's spam level. This section provides a description of these techniques.

Presence. Whenever we receive a voice call, whether we answer it or not depends on our state of mind. This state of mind can change depending on several factors such as location, the time, and our mood. To provide the VSD with parameters for determining their state-of-mind, end users can configure their phones with modes such as "do-not-disturb-me," "follow-me," and "911-emergency"-modes depending on their preferences and needs. One such example of integrating state-of-mind with filtering process is to synchronize the

system with the end user's calendar. The filtering process that takes place during this stage is based on static and dynamic rules configured by the end user.

Rate Limiting. Based on known traffic patterns, signatures can be used to detect the rate of incoming calls. For example, velocity and acceleration values (first and second order derivative) of the number of arriving calls from a given user/host/domain can be used as a detection mechanism. When the velocity/acceleration reaches a certain threshold, the drop rate can be updated through feedback control. As expected, the earlier we detect a change in the incoming pattern based on signatures, the earlier there will be a reduction in the spread of spam. Once spamming is identified, the filter can use Proportional Integral Control (PID), a feedback control, to reduce the velocity of spreading. This method of detection is useful not only in deterring spamming attacks but also in averting DOS attacks. Some preliminary investigation on the effectiveness of this detection method is presented in Dantu et al. [2004].

Black and White Lists. Most of the present-day spam filters conduct static checking of a set of signatures (a set of valid and invalid entities such as a user, host and domain). The spam filter uses these signatures to allow or block calls. *Whitelist* represents the set of entities from which the callee is always ready to receive calls. Similarly, *blacklist* represents the set of entities from which the callee prefers not to receive calls. Depending upon the callee's specifications, the calls with the specified entities will be allowed or denied calling. Such lists are customized, that is, each callee has the flexibility of specifying a set of whitelist and blacklist entities. The legitimate and spam entities in each of the callee's whitelist and blacklists will differ from other callees' lists, and thus, will not influence the call forwarding or blocking of other callees, that is, each end user is guaranteed of forwarded or denied calls based on a customized list.

The VSD constructs black- and whitelists using callee's feedback. If the callee's feedback for a forwarded call indicates spam, the VSD adds the call entities to the blacklist. Future calls from any entity on the blacklist are blocked at the VSD. On the other hand, if the callee responds with a legitimate-call feedback, the call entities are added to the callee's whitelist, and calls from them are forwarded to the callee.

Trust. Learning the caller's spam and legitimate behavior over time allows us to make many intelligent decisions regarding the call. This process of observing the caller's behavior constitutes the "trust" level the caller has built with the callee. *Trust* as such represents an abstract modeling of the caller's and the callee's past mutual behavior. This trust information can be used to classify an incoming call as spam or legitimate.

When the VSD needs to compute the trust of an incoming SIP voice call, it checks for previous trust information associated with the call-participating entities such as the call source (calling user, calling host, call-generating domain), participating proxies in routing with the help of fields such as "from," "to," "record route," and "via." The call's trust level is then computed using Bayesian inference techniques (see Section 4.3.1). If the call is forwarded to the callee, VSD updates the history of the caller to appropriately reflect the callee's feedback. At times, it is possible that, due to unavailability of previous

transactions, VSD cannot compute a trust value. In this case, we infer the caller's reputation from callee's neighbors.

Social Networks and Reputation. Social networks can be used to represent user relationships that can be derived along the network paths. These social networks can be used to infer the associated relations between the elements in the network. These relationships are transitive and transparent. If Alice is related to Bob and Bob is related to Charles, then with a reasonable degree of confidence, Charles can derive the trust information of Alice from Bob. With respect to a VoIP service user, the user's social network represents the associated and trusted neighbors from whom the user is willing to receive calls. While the trust is computed based on history, we derive reputation from trusted peers. The reputation of the call source can be inferred based on the previous experience of those trusted peers (Section 4.3.2) with that call source.

It is highly imperative for spam filters to be integrated with human behavioral aspects and principles to mimic the way humans' answer calls from wanted people. Applying these social notions of trust and reputation helps in identifying the social community and the relative closeness among the members of the community. Such information can then be used to improve the accuracy of identifying unwanted calls.

4.3 Trust and Reputation in Voice Calls

Formal models of trust have been proposed in security and in social sciences [Ray and Chakraborty 2004; Hepburn and Wright 2003; Orbaek and Palsberg 1997]. These papers, however, do not address the social notions of trust and reputation for solving real-time problems such as spam. The trust and reputation formalism presented here precisely addresses this problem. In particular, we attempt to address the following:

(1) Formalism structured on human intuitive behavior for detecting spam based on trust (direct) and reputation (indirect) relationships with the caller.
(2) A quantitative model based on the formalism that computes the number of spam calls required to move a caller from a whitelist to a blacklist and vice-versa.

For defining the quantitative model, we use Bayesian inference techniques to compute and update the trust and reputation relationships based on intuitive considerations. Variants of Bayesian estimation methodologies have been used in solving a multitude of problems relating to probabilistic reasoning and statistics. We believe that these Bayesian analysis and inference techniques would aid in automated and adaptive learning of spam behavior. This formalism for voice calls is integrated into VSD (Section 3). In A.1 of the Appendix, we present the terminology adopted for the formalism. In Section 4.3.1, we present the trust formalism that describes a model for computing and updating trust based on callee's feedback. Section 4.3.2 presents a reputation formalism for inferring and updating reputation based on callee's feedback. Section 4.3.3 explains the integration of the above models of trust and reputation for computing the spam probability of the call.

4.3.1 *Trust.* Trust has been traditionally used in solving the problem of authentication in application areas such as ad-hoc and peer-to-peer systems. Social notions of trust can be used in inferring the spam behavior of voice calls. To begin, we define trust in context of analyzing voice calls by modifying the definition given in Wang and Vassileva [2003b] as *a callee's belief in caller's capabilities, honesty and reliability based on his/her own direct experiences*. Trust refers to caller's capability, honesty, and reliability in making legitimate calls to the callee. This trust of the incoming call is based on the trust of the individual participants of the call and the callee's experiences towards those call participants.

Property 1. Trust level of a voice call depends on the call participants.

The trust T of the incoming SIP voice call depends on the trust of individual call participants. A call participant can be a user, a host, a domain, or an intermediate proxy.

Property 2. Trust is derived from the caller's past behavior.

Trust for a call participant is accrued over a period of time based on its past behavior. For each call participant i, we denote a call set $C_i = \{N_{i,s}, N_{i,v}\}$ the spaminess and legitimateness of participant i. The spaminess $N_{i,s}$ represents the total number of past spam calls and legitimateness $N_{i,v}$ represents the total number of past legitimate calls from the call participant. The higher a call participant's spaminess, the higher are the chances that the call having this call participant will be filtered. Similarly, the higher the legitimateness, the higher the chances that VSD will forward the call having this call participant to the callee. The initial values of $N_{i,s}$ and $N_{i,v}$ of a call participant i are defined by $N_{i,s}, N_{i,v} = 1$, when VSD has no history for the call participant i. Alternatively, with available history for the call participant with respect to the callee, the VSD increments $N_{i,s}$ for every spam call and $N_{i,v}$ for every legitimate call to the callee. This spaminess and legitimateness of individual call participants helps in computing the overall trust of the incoming call.

The trust of the incoming call is inferred from $D = f(C, N_S, N_V)$ where N_S *and* N_V represents the total number of spam and legitimate calls processed by VSD. C represents the set of call sets of all participants, that is, $C = \{C_1, C_2, \ldots, C_n\}$ where n is the number of call participants. D is defined as the *distrust* of the call. The higher the distrust of the call, the lower is the trust level associated with it. $D \in [0\ 1]$, that is, the distrust of the call lies in the range [0 1]. The distrust D of the incoming call is dependent on the spaminess and legitimateness of all the call participants and is computed using Bayesian Analysis. This is represented by:

$$D = f(C, N_S, N_V)$$

that is,

$$D = f(\{N_{1,S}, N_{1,V}\}, \{N_{2,S}, N_{2,V}\}, \{N_{3,S}, N_{3,V}\}, \ldots, \{N_{n,S}, N_{n,V}\}, N_S, N_V),$$

where the function "f" is defined as (detailed probabilistic model is explained in A.2 of the Appendix)

$$D = \frac{\left(\frac{\sum_{i=1}^{n} N_{i,s}}{\sum_{i=1}^{n}(N_{i,s}+N_{i,v})}\right)\prod_{i=1}^{n}\frac{N_{i,s}}{N_{i,s}+N_{i,v}}}{\left(\frac{\sum_{i=1}^{n} N_{i,s}}{\sum_{i=1}^{n}(N_{i,s}+N_{i,v})}\right)\prod_{i=1}^{n}\frac{N_{i,s}}{N_{i,s}+N_{i,v}} + \left(\frac{\sum_{i=1}^{n} N_{i,v}}{\sum_{i=1}^{n}(N_{i,s}+N_{i,v})}\right)\prod_{i=1}^{n}\frac{N_{i,v}}{N_{i,s}+N_{i,v}}} \quad (1)$$

Higher the value of D, higher is the chance that the call is going to be filtered. The simple Bayesian equation shown above helps us in computing the distrust of the incoming call. Though we have carried out a treatment of simple Bayesian analysis for processing calls, we believe that the results are valid even for different variants of Bayesian analysis and techniques.

Definition 1. Trust level T is a direct measure of distrust and is equal to 1-D.

The distrust D given in Eq. (1) helps in computing the value of trust for the incoming call. This computation is a direct measure of distrust D and is equal to the value 1-D.

AXIOM 1. *Callers can be grouped based on their calling patterns.*

In real world, we remember people such as our friends, family members, neighbors, and unwanted callers who can be grouped into white, grey and black lists. The membership of these lists varies depending upon our mood, past experience, current needs and distrust. For example, we can assign some ranges of distrust levels to these lists as follows:

$$D = \begin{cases} 0 - 0.01 & \text{White list} \\ 0.01 - 0.99 & \text{Grey list} \\ 0.99 - 1.0 & \text{Black list.} \end{cases}$$

A caller in a callee's blacklist can be a user who has had a spam behavior for a long time such that the caller's distrust was as high as 0.99. However, a reasonable number of legitimate calls from the caller can decrease this distrust value. If the distrust falls below a threshold of 0.01, then the caller can be added to the callee's whitelist. In addition, the above scale can also be used when quarantining incoming calls. For example, with a distrust value more than 0.99, the incoming call can be directly blocked, and for distrust less than 0.01, the call can be directly forwarded. The computed distrust (Property 2) after integrating with reputation inference (Section 4.3.2) is compared with a predetermined threshold configured by the callee and a decision can be made whether to forward the call to the callee or to filter it. If the call is filtered, then it can be sent to the voicemail box or completely blocked, depending upon the callee's customized options. Alternatively, if the call is forwarded to the callee, then the callee can answer the call and give feedback to the VSD. The callee responding with a spam or legitimate call feedback constitutes the explicit feedback. However, another type of feedback can be inferred from the callee's calling behavior. It is the implicit feedback available to the VSD when the callee makes outgoing calls (here, the role of the callee changes to a caller). In this case, the VSD updates the history of the called parties with respect to him.

There is a marked difference in the way trust is updated based on implicit and explicit feedback and is explained in the following property.

Property 3. Trust can be derived from incoming as well as outgoing calls. Every wanted or legitimate incoming call translates to an additive increase in trust whereas an outgoing call results in an exponential increase.

For a given callee, *trust* is a function of both *incoming* and *outgoing* calls. It is normal that the trust we have towards people we call is greater than the *trust* we attribute to people who call us. The trust we bestow on people who receive calls from us increases exponentially with every *outgoing* call whereas the trust attributed to people who call us increases additively. We incorporate this behavior into VSD as follows:

For a *positive feedback* from the callee for the present call, the distrust D for the next call from the caller would be updated by:

$$D = f(\{N_{1,s}, N_{1,v}+1\}, \{N_{2,s}, N_{2,v}+1\}, \{N_{3,s}, N_{3,v}+1\}, \ldots, \{N_{n,s}, N_{n,v}+1\}, N_S, N_V+1).$$

If the call is *spam*, as specified by the callee, distrust D is updated by

$$D = f(\{N_{1,s}+1, N_{1,v}\}, \{N_{2,s}+1, N_{2,v}\}, \{N_{3,s}+1, N_{3,v}\}, \ldots, \{N_{n,s}+1, N_{n,v}\}, N_S+1, N_V).$$

But, for an *outgoing call*, trust is increased exponentially and is represented by

$$D = f(\{N_{1,s}, N_{1,v}\underline{|e^{k_1}|}\}, \{N_{2,s}, N_{2,v}\underline{|e^{k_2}|}\}, \{N_{3,s}, N_{3,v}\underline{|e^{k_3}|}\}, \ldots, \{N_{n,s}, N_{n,v}\underline{|e^{k_n}|}\}, N_S, N_V).$$

We believe that $k_i \geq 0$ and is proportional to individual trust (T_i) of the call participant i for $i = 1 \cdot \cdot n$, that is, the amount of exponential increase is in the order of trust of the respective call participant.

Therefore,k_i is *proportional* to T_i
that is, $k_i \propto T_i$ for $i = 1 \cdot \cdot n$

For defining the trust (T_i) of individual call participant i, we compute the distrust (D_i) of the call participant. Trust is then inferred directly from the distrust value and can be safely assumed to be equal to $1 - D_i$. The distrust D_i for call participant i (variant of A.2 of the Appendix for one random variable) is given by

$$D_i = \frac{\left(\frac{N_{i,s}}{N_S}\right)\left(\frac{N_{i,s}}{N_{i,s}+N_{i,v}}\right)}{\left(\frac{N_{i,s}}{N_S}\right)\left(\frac{N_{i,s}}{N_{i,s}+N_{i,v}}\right) + \left(\frac{N_{i,v}}{N_V}\right)\left(\frac{N_{i,v}}{N_{i,s}+N_{i,v}}\right)} \tag{2}$$

Computing individual distrust for each call participant helps to identify the amount of spam behavior associated with that call participant. In addition, this computation assists in reducing false alarms (i.e., the total number of false positives and false negatives). Therefore, for $i = 1 \cdot \cdot n$, the lower the value of D_i from Eq. (2), the higher is the value of k_i and, therefore, the higher the increase in trust level and vice-versa. Trust can be updated by computing the distrust (as shown in Eq. (2)) for every incoming call passing through the VSD.

Based on the above constructs, we can derive a quantitative model using Bayesian estimation that computes the number of spam or legitimate calls required for moving the caller among the lists defined in Axiom 1. To achieve this,

the current behavior of a call participant, that is, the spaminess and legitimateness of a call participant based on its past behavior must be computed. We, as humans, do this in our daily life as well. When receiving a voice call, we check the caller-id of the incoming call and intuitively estimate the likelihood of the call being spam based on the caller's trustworthiness and past behavior.

LEMMA 1. *A participant's spaminess can be inferred from its past calls and current distrust.*

Let D_i be the distrust of a call participant "i" for its past total calls $N_{i,B}$ where $N_{i,B} = N_{i,s} + N_{i,v}$. Let $N_{i,s}$ and $N_{i,v}$ represent the spaminess and legitimateness associated with the call participant; then,

$$N_{i,s} = h_1(N_{i,B}, D_i, N_S, N_V)$$
$$N_{i,V} = h_2(N_{i,B}, D_i, N_S, N_V),$$

where the functions h_1 and h_2 are derived by solving the two equations

$$N_{i,B} = N_{i,s} + N_{i,v} \tag{3}$$

and

$$D_i = \frac{\left(\frac{N_{i,s}}{N_S}\right)\left(\frac{N_{i,s}}{N_{i,s}+N_{i,v}}\right)}{\left(\frac{N_{i,s}}{N_S}\right)\left(\frac{N_{i,s}}{N_{i,s}+N_{i,v}}\right) + \left(\frac{N_{i,v}}{N_V}\right)\left(\frac{N_{i,v}}{N_{i,s}+N_{i,v}}\right)}$$

from Eq. (2)

From above, we have

$$\begin{aligned}
D_i &= \frac{\left(\frac{N_{i,s}}{N_S}\right)(N_{i,s})}{\left(\frac{N_{i,s}}{N_S}\right)(N_{i,s}) + \left(\frac{N_{i,v}}{N_V}\right)(N_{i,v})} \\
&\Rightarrow D_i = \frac{N_V(N_{i,s})^2}{N_V(N_{i,s})^2 + N_S(N_{i,v})^2} \\
&\Rightarrow (N_{i,v})^2 = \frac{N_V(N_{i,s})^2(1-D_i)}{N_s D_i} \Rightarrow \frac{(N_{i,v})^2}{(N_{i,s})^2} = \frac{N_V(1-D_i)}{N_S D_i} \\
&\Rightarrow \frac{N_{i,v}^2}{(N_{i,B}-N_{i,v})^2} = \frac{N_V(1-D_i)}{N_S D_i} \; from(3) \; \Rightarrow \frac{(N_{i,B}-N_{i,v})^2}{(N_{i,v})^2} = \frac{N_S D_i}{N_V(1-D_i)} \\
&\Rightarrow \frac{N_{i,B}}{N_{i,v}} = 1 + \sqrt{\frac{N_S D_i}{N_V(1-D_i)}} \Rightarrow N_{i,v} = \frac{N_{i,B}}{1+\sqrt{\frac{N_S D_i}{N_V(1-D_i)}}}
\end{aligned} \tag{4}$$

Therefore,

$$\begin{aligned}
N_{i,s} &= N_{i,B} - N_{i,v} \Rightarrow N_{i,s} = N_{i,B} - \frac{N_{i,B}}{1+\sqrt{\frac{N_S D_i}{N_V(1-D_i)}}} \\
&\Rightarrow N_{i,s} = \frac{N_{i,B}}{1+\sqrt{\frac{N_V(1-D_i)}{N_S D_i}}}
\end{aligned} \tag{5}$$

Therefore, given distrust D_i and past number of calls $N_{i,B}$ for a call participant i, we can find $N_{i,s}$ and $N_{i,v,}$ that is, the call participant's spaminess and legitimateness. Deriving $N_{i,s}$ and $N_{i,v}$ for a given distrust and total past calls helps us to compute the number of spam or legitimate calls required to move a caller from one list (e.g., whitelist) to another (e.g., blacklist) defined in Axiom 1 as shown in Lemma 2.

LEMMA 2. *The number of calls* (N_{CF}) *needed to identify spam depends on the caller's distrust and the callee's tolerance.*

In reality, it is highly likely that people will change their behavior such as legitimate callers generating unwanted calls or unsolicited callers making legitimate calls. In this context, we derived a quantitative model that computes the number of spam calls required to categorize a caller to a blacklist. This is more necessary in VoIP than in e-mail. In the case of e-mail, it would not be a serious nuisance to the end user if the spam filter doesn't stop spam emails (false negatives). But, spam calls become a greater nuisance in case of VoIP because the end user must answer the call in real time. For deriving N_{CF}, assume that N'_S and N'_V represent the total number of spam and legitimate calls processed by the VSD. In this lemma, consider that three call participants—calling user, calling host, and call-generating domain—are used to compute an incoming call's distrust value. Assume

Spaminess of user $(N_{1,s}) = N'_{1,s}$, and legitimateness of the user $(N_{1,v}) = N'_{1,v}$;

Similarly, spaminess of host $(N_{2,s}) = N'_{2,s}$, and legitimateness of host $(N_{2,v}) = N'_{2,v}$.

Spaminess of domain $(N_{3,s}) = N'_{3,s}$, and legitimateness of domain $(N_{3,v}) = N'_{3,v}$.

Therefore, the current distrust (D_C) based on the spaminess and legitimateness of the three call participants of the incoming call derived using Eq. (1) is given by,

$$D_C = \frac{\left(\frac{\sum_{i=1}^{3} N'_{i,s}}{\sum_{i=1}^{3}(N'_{i,s}+N'_{i,v})}\right)\prod_{i=1}^{3}\frac{N'_{i,s}}{N'_{i,s}+N'_{i,v}}}{\left(\frac{\sum_{i=1}^{3} N'_{i,s}}{\sum_{i=1}^{3}(N'_{i,s}+N'_{i,v})}\right)\prod_{i=1}^{3}\frac{N'_{i,s}}{N'_{i,s}+N'_{i,v}} + \left(\frac{\sum_{i=1}^{3} N'_{i,v}}{\sum_{i=1}^{3}(N'_{i,s}+N'_{i,v})}\right)\prod_{i=1}^{3}\frac{N'_{i,v}}{N'_{i,s}+N'_{i,v}}}$$

$$\Rightarrow D_C = \frac{\left(\sum_{i=1}^{3} N'_{i,s}\right)\prod_{i=1}^{3} N'_{i,s}}{\left(\sum_{i=1}^{3} N'_{i,s}\right)\prod_{i=1}^{3} N'_{i,s} + \left(\sum_{i=1}^{3} N'_{i,v}\right)\prod_{i=1}^{3} N'_{i,v}}$$

$$\Rightarrow D_C = \frac{(N'_{1,s}+N'_{2,s}+N'_{3,s})N'_{1,s}N'_{2,s}N'_{3,s}}{(N'_{1,s}+N'_{2,s}+N'_{3,s})N'_{1,s}N'_{2,s}N'_{3,s}+(N'_{1,v}+N'_{2,v}+N'_{3,v})N'_{1,v}N'_{2,v}N'_{3,v}}$$

From the above equation, we have

$$\frac{D_C}{1-D_C} = \frac{(N'_{1,s}+N'_{2,s}+N'_{3,s})N'_{1,s}N'_{2,s}N'_{3,s}}{(N'_{1,v}+N'_{2,v}+N'_{3,v})N'_{1,v}N'_{2,v}N'_{3,v}} \tag{6}$$

Now, assume that there were N_{CF} number of spam calls from the call participants (i.e., the calling user, calling host, and call-generating domain) after

which the calls are filtered. Therefore, because of linearly updating the history of each call participant based on feedback from the callee, the history of the three call participants is given by

Spaminess of user $(N_{1,s}) = N'_{1,s} + N_{CF}$ and legitimateness of user $(N_{1,v}) = N'_{1,v}$;

Spaminess of host $(N_{2,s}) = N'_{2,s} + N_{CF}$ and legitimateness of host $(N_{2,V}) = N'_{2,v}$;

Spaminess of domain $(N_{3,s}) = N'_{3,s} + N_{CF}$ and legitimateness of domain $(N_{3,v}) = N'_{3,v}$;

Total spam calls processed by VSD = $N'_S + N_{CF}$ and total number of legitimate calls processed by VSD = N'_V. Therefore, the final distrust level (D_F) after N_{CF} number of spam calls from the 3 call participants, is given by

$$D_F = \frac{\left(\frac{\sum_{i=1}^{3}(N'_{i,s}+N_{CF})}{\sum_{i=1}^{3}(N'_{i,s}+N_{CF}+N'_{i,v})}\right)\prod_{i=1}^{3}\frac{N'_{i,s}+N_{CF}}{N'_{i,s}+N_{CF}+N'_{i,v}}}{\left(\frac{\sum_{i=1}^{3}N'_{i,s}+N_{CF}}{\sum_{i=1}^{3}(N'_{i,s}+N_{CF}+N'_{i,v})}\right)\prod_{i=1}^{3}\frac{N'_{i,s}+N_{CF}}{N'_{i,s}+N_{CF}+N'_{i,v}} + \left(\frac{\sum_{i=1}^{3}N'_{i,v}}{\sum_{i=1}^{3}(N'_{i,s}+N_{CF}+N'_{i,v})}\right)\prod_{i=1}^{3}\frac{N'_{i,v}}{N'_{i,s}+N_{CF}+N'_{i,v}}}$$

$$\Rightarrow D_F = \frac{\begin{array}{c}(N'_{1,s}+N'_{2,s}+N'_{3,s}+3N_{CF})(N'_{1,s}+N_{CF})\\(N'_{2,s}+N_{CF})(N'_{3,s}+N_{CF})\end{array}}{\begin{array}{c}(N'_{1,s}+N'_{2,s}+N'_{3,s}+3N_{CF})(N'_{1,s}+N_{CF})(N'_{2,s}+N_{CF})\\(N'_{3,s}+N_{CF}) + (N'_{1,v}+N'_{2,v}+N'_{3,v})N'_{1,v}N'_{2,v}N'_{3,v}\end{array}}$$

$$\Rightarrow \frac{(N'_{1,s}+N'_{2,s}+N'_{3,s}+3N_{CF})(N'_{1,s}+N_{CF})(N'_{2,s}+N_{CF})(N'_{3,s}+N_{CF})}{(N'_{1,v}+N'_{2,v}+N'_{3,v})N'_{1,v}N'_{2,v}N'_{3,v}} = \frac{D_F}{1-D_F}$$

Using Eq. (6), we have

$$\Rightarrow \frac{(N'_{1,s}+N'_{2,s}+N'_{3,s}+3N_{CF})(N'_{1,s}+N_{CF})(N'_{2,s}+N_{CF})(N'_{3,s}+N_{CF})}{(N'_{1,s}+N'_{2,s}+N'_{3,s})N'_{1,s}N'_{2,s}N'_{3,s}} = \frac{D_F(1-D_C)}{(1-D_F)D_C} \tag{7}$$

For solving Eq. (7) for N_{CF}, we require the values of $N'_{i,s}$ and $N'_{i,v}$ for $i = 1..3$. $N'_{i,s}$ and $N'_{i,v}$ are dependent on the distrust of participant "i" and its past calls $N'_{i,B}$ for $i = 1..3$ as presented in Lemma 1. Substituting $N'_{i,s}$ and $N'_{i,v}$ for $i = 1..3$ with the current distrust D_C and final distrust D_F in Eq. (7), the number of calls N_{CF} required to move from distrust D_C to distrust D_F can be computed. Using Eq. (7), the number of spam calls required to move a caller from a list (e.g., whitelist) to another list (e.g., blacklist) can be computed by assuming the values of current distrust (D_C) and final distrust (D_F) based on the values defined for the lists in Axiom 1.

COROLLARY 1. *The number of spam calls* (N_{CF}) *required by VSD to identify a new spammer is* 3.

Equation (7) is used for computing the number of spam calls required by the VSD for moving the distrust of an incoming call from D_C to distrust D_F. For a new spammer from a new host and domain, the spaminess and legitimateness

for each call participant in the call is equal to 1, that is, $N'_{1,s} = N'_{1,v} = N'_{2,s} = N'_{2,v} = N'_{3,s} = N'_{3,v} = 1$ (Property 2). These values if substituted in Eq. (1) would result in an initial distrust value $D_C = 0.5$. Therefore, substituting above values, the value of D_C, and threshold $T = 0.99$ (therefore $D_F = 0.99$) to directly get filtered in Eq. (7), we get a value of

$$N_{CF} \approx 3, \tag{8}$$

that is, VSD takes 3 spam calls to move from initial probability of 0.5 to the threshold probability D_F = 0.99. This is experimentally validated in Section 5.2.1. In practice, the value of N_{CF} depends on the threshold value.

Note 1. In this lemma, we assume that the number of spam calls from the host and domain are same as the number of spam calls from the user (N_{CF}). However, it is quite possible that there might be different user accounts on the same IP phone and many hosts in the same domain. In this situation, depending upon the spam behavior of other users, the spam histories of the host and of the domain change. Therefore, at any instant, the number of calls required to cross the threshold would be dependent on the spaminess and legitimateness of each of the call participants as is modeled using Eq. (1).

Note 2. The quantitative model computes the number of spam calls required to categorize a caller into blacklist by increasing the call participants' spaminess. A similar model can be used to compute the number of legitimate calls to categorize the caller into a whitelist. However, in this case, the spaminess remains the same and legitimacy of the call participants change, that is, after N_{CF} number of legitimate calls, the spaminess ($N_{i,S}$) remains $N'_{i,s}$, but the legitimateness ($N_{i,v}$) changes to $N'_{i,v}+N_{CF}$ for each call participant i for $i = 1\cdot\cdot3$.

Note 3. The number of call participants for analysis can be extended from the three participants of calling user, calling host, and call-generating domain to include other call participants such as the source and intermediate proxies. In this case, to categorize a caller into blacklist, N_{CF} would be a function of the individual spaminess of all the "n" call participants, that is, given spaminess $N_{i,s}$ for $i = 1 \cdot\cdot n$ and the distrusts D_C and D_F, the number of spam calls N_{CF} can be computed. Similarly, to categorize a caller into a whitelist, N_{CF} would be a function of the individual legitimateness of all the "n" call participants, that is, given legitimateness $N_{i,v}$ for $i = 1\cdot\cdot n$ and the distrusts D_C and D_F, the number of legitimate calls N_{CF} can be computed.

Note 4. Corollary 1 is a specific case for a new spammer who does not have history of calling any of the users inside the $\text{VSD}_{\text{domain}}$. Due to this, we initialize the spaminess and legitimateness of the call participants from this new caller to be 1 and derive a value of $N_{CF} = 3$. However, for calls from other callers that have a history of calling users inside the $\text{VSD}_{\text{domain}}$, the value of N_{CF} depends on the previous spaminess and legitimateness of the incoming call's participants. Higher spaminess compared to legitimateness of the call participants results in a value of N_{CF} less than 3, and higher legitimateness compared to spaminess of call participants results in a value of N_{CF} greater than 3. To generalize, for a call from new spammer with n call participants, the spaminess and legitimateness

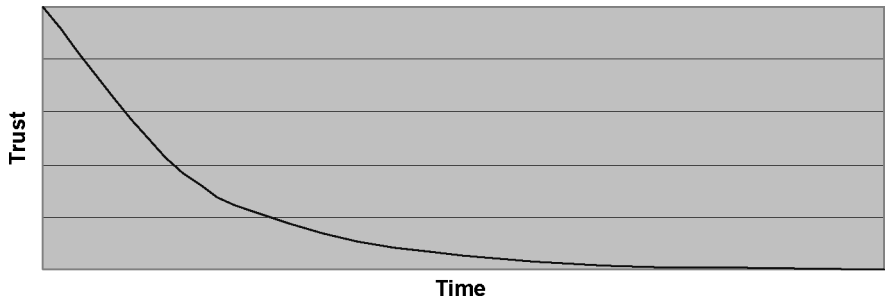

Fig. 3. Trust fades with time. In the absence of transactions, trust decreases exponentially in the order of elapsed time.

of each call participant can be initialized to a value 1 for computing the value of N_{CF}.

The above quantitative model is derived based on an assumption that there are spam or legitimate calls from the caller to the callee. However, it is possible that a callee does not receive a call from the caller for a long time. In this case, the trust level of the caller fades over time.

Property 4. We forget bad and good experience over time, and as a result, in the absence of any transactions, trust fades.

Trust is accrued over a period of time. This accrued trust is a representation of the caller's past behavior. But, in absence of calls from the caller over a period of time, the trust decreases, that is, trust fades with time. This fading of trust is exponential (Palla and Dantu [2006]). If the last transaction with the caller was at time t_p in the interval $\{t_1, t_n\}$ such that $t_1,\ldots.t_p,\ldots.t_n$, then the trust value will decay over the time period $\Delta t = t_n - t_p$. This can be represented by $T_{i,n} = T_{i,p} \exp(-\Delta t)$ where $T_{i,p}$ and $T_{i,n}$ represent the trust for a given call participant "i" for $i = 1 \cdot \cdot n$, at time periods t_p and t_n respectively. This fading of trust over a period of time for a caller is as shown in Figure 3.

All the above notions of trust can be applied only when there is an available caller history for the incoming call. But, in everyday life, we receive calls from individuals who are calling for the first time (e.g., unknown callers - strangers). We lack a prior calling history for them. In this case, we must rely on reputation or on recommendations based on word of mouth.

4.3.2 *Reputation.* It is a human tendency to rely on the opinions of trusted people regarding an individual's trustworthiness (usually referred to as the individual's *reputation*) in addition to one's own experience. Knowing the individual's reputation becomes even more necessary when we have no previous experience with that individual. Inferring reputation for detecting the spam behavior of a call is useful particularly in cases when neighbors have first-hand experience with the caller. Here we present a model for inferring reputation and updating it based on callee's feedback. But, first, we define reputation of a call participant as a notion or report of its propensity to fulfill the trust placed in it (during a particular situation); its reputation is created through feedback from

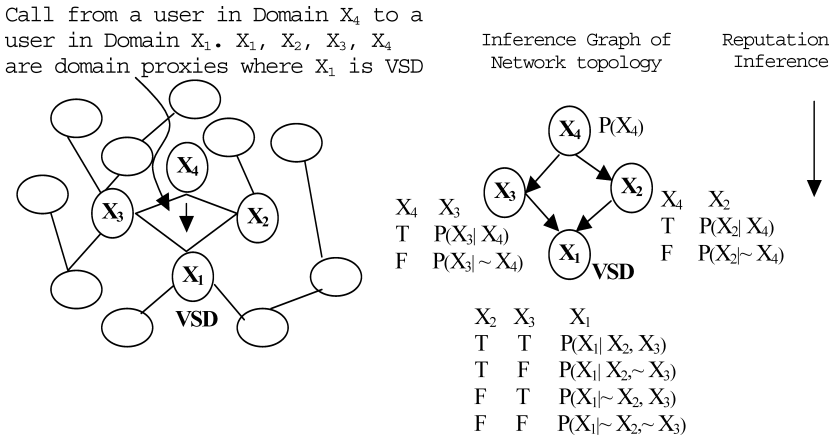

Fig. 4. Reputation inference for a call from domain X_4 to domain X_1. The proxy topology for reputation takes into account the interconnection among domain proxies that are in all the possible paths from the caller's domain proxy to VSD. This topology can then be used for propagating and updating reputation information using Bayesian Networks based on an observed evidence of spam based on callee's feedback.

individuals who have previously interacted with the call participant (Goecks and Mynatt [2002]).

The reputation of a call participant is inferred based on the recommendations of the neighbors of the callee (e.g., other employees in the enterprise) as given in Ono and Schulzrinne [2005]. These neighbors can in turn poll their neighbors for the call participant's reputation. This reputation mechanism can be integrated into the functionality of VSD. For this integration, instead of the actual callee seeking the recommendations regarding each call participant, the VSD seeks recommendations about the caller's domain proxy from its own neighboring domain proxies. The neighboring proxies, in turn, seek the recommendation of their neighboring proxies until the caller's domain proxy is reached. VSD then infers the reputation of the caller's domain proxy based on these recommendations using Bayesian Networks inference techniques. For this, with respect to a domain proxy, a graph can be generated using the neighboring proxies that can be used in deriving the caller's domain reputation. For example, consider an example ring proxy network topology [Lancaster 2003] graph given in Figure 4.

For the given topology graph in Figure 4, reputation is inferred using Bayesian networks. For a call from a domain (X_2, X_3 or X_4) to a user inside X_1 ($\text{VSD}_{\text{domain}}$), the reputation of the domain can be updated by feedback from the end user, that is, the evidence is propagated throughout the Bayesian network (detailed probabilistic model is explained in A.3 of the Appendix). For a call from domain X_4 to domain X_1, the reputation of domain X_4 can be inferred by computing $P(X_1|X_4)$, that is, the posterior probability of X_1 given an event that a call has been generated at X_4.

$$\begin{aligned} P(X_1|X_4) &= P(X_1, X_2|X_4) + P(X_1, \sim X_2|X_4) \\ &= P(X_1|X_2)P(X_2|X_4) + P(X_1|\sim X_2)P(\sim X_2|X_4) \end{aligned} \tag{9}$$

where

$$\begin{aligned} P(X_1|X_2) &= P(X_1, X_3|X_2) + P(X_1, \sim X_3|X_2) \\ &= P(X_1|X_2, X_3)P(X_3) + P(X_1|X_2 \sim X_3)P(\sim X_3) \end{aligned} \quad (10)$$

and

$$P(X_1| \sim X_2) = P(X_1, X_3| \sim X_2) + P(X_1, \sim X_3| \sim X_2) = P(X_1| \sim X_2, X_3)P(X_3) + P(X_1| \sim X_2, \sim X_3)P(\sim X_3) \quad (11)$$

$$P(X_3) = P(X_{3,}X_4) + P(X_3, \sim X_4) = P(X_3|X_4)P(X_4) + P(X_3| \sim X_4)P(\sim X_4) \quad (12)$$

Solving Eqs. (9)–(12) gives the reputation $P(X_1|X_4)$of domain proxy X_4. The inferred reputation of caller's domain is then updated based on callee's feedback. The reputation for the caller domain that is inferred can then be used either for increasing or decreasing the trust level (see Section 4.3.3) of the incoming call.

Property 5. Trust and reputation levels increase additively and decrease multiplicatively.

In our daily life, we slowly gain trust, but we develop distrust quickly. We model this intuitive behavior in updating the trust and the reputation values. However, previously in this article, we have adopted a linear model in updating trust (or distrust). Here, we present an alternative model for updating trust. The type of update model to be used depends on the sensitivity of the underlying application.

For a *legitimate-call feedback* from a callee, the distrust D is decreased as shown below (increasing the legitimateness decreases the distrust)

$$D = f(\{N_{1,s}, N_{1,v}+T_1\}, \{N_{2,s}, N_{2,v}+T_2\}, \{N_{3,s}, N_{3,v}+T_3\}, \ldots, \{N_{n,s}, N_{n,v}+T_n\}, N_S, N_V+1).$$

Similarly, reputation is additively increased for good behavior. For the graph topology in Figure 4, the parameters for each node are updated additively for a *legitimate* call. For example, for a *legitimate-call* feedback from callee for a call from domain X_4 to domain X_1, the parameters of each node are updated as follows:

$$\text{Node } X_2 : P(X_2|X_4) = P(X_2|X_4) + r_1 \quad P(X_2| \sim X_4) = P(X_2| \sim X_4) - r_1$$

$$\text{Node } X_3 : P(X_3|X_4) = P(X_3|X_4) + r_1 \quad P(X_3| \sim X_4) = P(X_3| \sim X_4) - r_1$$

$$\text{Node} X_1 : P(X_1|X_2, X_3) = P(X_1|X_2, X_3) + \frac{P(X_1|X_2)}{S_{\text{inf}}}s_1 + \frac{P(X_1|X_3)}{S_{\text{inf}}}s_1$$

$$P(X_1|X_{2,} \sim X_3) = P(X_1|X_2, \sim X_3) + \frac{P(X_1|X_2)}{S_{\text{inf}}}s_1 - \frac{P(X_1|X_3)}{S_{\text{inf}}}s_1,$$

$$P(X_1| \sim X_2, X_3) = P(X_1| \sim X_2, X_3) - \frac{P(X_1|X_2)}{S_{\text{inf}}}s_1 + \frac{P(X_1|X_3)}{S_{\text{inf}}}s_1$$

$$P(X_1| \sim X_{2,} \sim X_3) = P(X_1| \sim X_2, \sim X_3) - \frac{P(X_1|X_2)}{S_{\text{inf}}}s_1 - \frac{P(X_1|X_3)}{S_{\text{inf}}}s_1,$$

where r_1 and s_1 are constants and $S_{inf} = P(X_1|\, X_2) + P(X_1|\, X_3)$.

Alternatively, for a *spam* call, both the trust and the reputation levels decrease multiplicatively. this is represented as follows:

$$D = f(\{N_{1,s} + J_1 D_1, N_{1,v}\}, \{N_{2,s} + J_2 D_2, N_{2,v}\}, \{N_{3,s} + J_3 D_3, N_{3,v}\}, \ldots, \{N_{n,s} + J_n D_n, N_{n,v}\}, N_S + 1, N_V)$$

where D_i is the associated distrust and J_i is multiplicative constant for updating distrust for a call participant i for $i = 1 \cdot \cdot n$. In the basic case, $J_i D_i = 1$ for $i = 1 \cdot \cdot n$ for experimenting with a linear increase in distrust.

Similarly for *reputation*, the parameters for each of the nodes in the graph topology would be updated for a *spam-call* feedback from the callee as follows:

$$\text{Node } X_2 : P(X_2|X_4) = P(X_2|X_4) - r_2 \quad P(X_2| \sim X_4) = P(X_2| \sim X_4) + r_2$$

$$\text{Node } X_3 : P(X_3|X_4) = P(X_3|X_4) - r_2 \quad P(X_3| \sim X_4) = P(X_3| \sim X_4) + r_2$$

$$\text{Node } X_1 : P(X_1|X_{2,}X_3) = P(X_1|X_2, X_3) - \frac{P(X_1|X_2)}{S_{\text{inf}}} s_2 - \frac{P(X_1|X_3)}{S_{\text{inf}}} s_2$$

$$P(X_1|X_{2,} \sim X_3) = P(X_1|X_2, \sim X_3) - \frac{P(X_1|X_2)}{S_{\text{inf}}} s_2 + \frac{P(X_1|X_3)}{S_{\text{inf}}} s_2$$

$$P(X_1| \sim X_2, X_3) = P(X_1| \sim X_2, X_3) + \frac{P(X_1|X_2)}{S_{\text{inf}}} s_2 - \frac{P(X_1|X_3)}{S_{\text{inf}}} s_2$$

$$P(X_1| \sim X_{2,} \sim X_3) = P(X_1| \sim X_2, \sim X_3) + \frac{P(X_1|X_2)}{S_{\text{inf}}} s_2 + \frac{P(X_1|X_3)}{S_{\text{inf}}} s_2,$$

where $S_{\text{inf}} = P(X_1|\, X_2) + P(X_1|\, X_3).r_2$, s_2 are constants such that $r_2 > l_1 r_1$ and $s_2 > l_2 s_1$ and $l_1, l_2 > 1$. The two constants l_1 and $l_2$are the multiplicative constants and are configured based on callee's preferences. The other constants r_1, r_2, s_1, s_2 are configuration parameters of VSD and can be initialized based on criteria such as the maximum number of calls that can be allowed from a spam domain and spam host. The updated values for the reputation parameters are in turn substituted in Eqs. (9)–(12) to result in a new set of updated reputation values for the nodes X_2 , X_3, and X_4 (represented by $P(X_1|\, X_2)$, $P(X_1|\, X_3)$ and $P(X_1|\, X_4)$ respectively). For a given set of initial or prior probabilities for the topology graph nodes representing the reputation of those domains, and for a spam call from domain X_4 to domain X_1, the Bayesian inference calculations shown above would decrease the reputation for X_2, X_3 and X_4 proxies and increase the reputation for a legitimate call for the same domain proxies. For every incoming call, this adaptive update of reputation is derived for all the domains in the probable path from the source domain proxy to VSD.

Property 6. With no prior experience, we rely on reputation. After multiple transactions with the caller, trust takes precedence and the influence of reputation decreases.

Many a times, trust and reputation are used to represent human belief. Trust represents a caller's past behavior whereas reputation signifies social status. While trust is computed, reputation is derived. Figure 5 presents a trust-and-reputation-influence plot based on human intuitive behavior in estimating the belief we place in individuals.

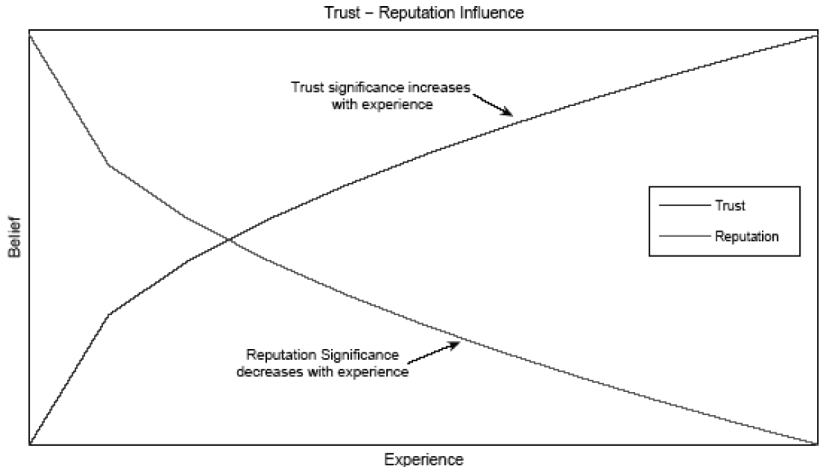

Fig. 5. Real life influence of trust and reputation. With no previous experience one relies mostly on reputation or recommendations. With increasing experience, the influence of trust (developed using experience) increases and that of the reputation decreases.

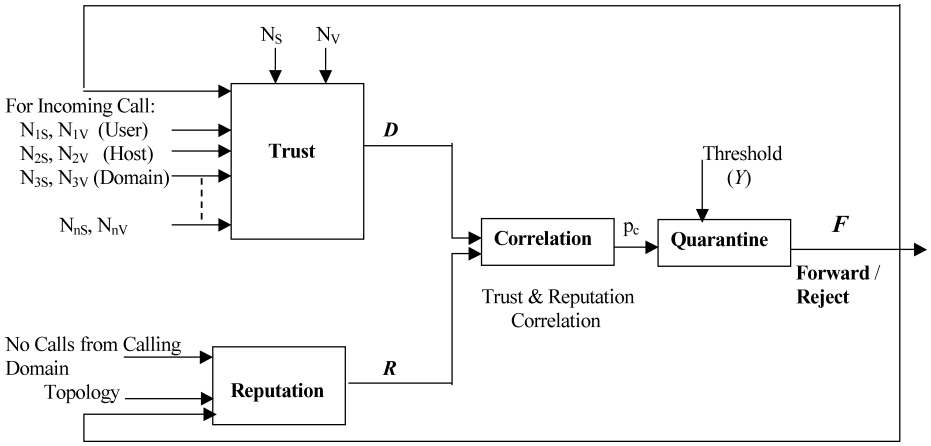

Fig. 6. Integration of trust and reputation. The trust is either increased or decreased based on reputation of caller's domain. The collective inference of these two stages results in a decision as to whether forward or quarantine the call

With no available history or experience, we rely mostly on the caller's reputation. Once we start receiving calls from the caller, trust would have more influence than reputation. This is particularly useful when our main goal is to define customized filters as the needs and perceptions of people change.

4.3.3 *Integrating Trust and Reputation.* The trust and reputation models defined in Section 4.3.1 and Section 4.3.2 are integrated as functional working modules in our voice-spam-filter analysis as shown in Figure 6.

For an incoming call, the spam level of the call can be computed using distrust D of the call by taking into account the spam and legitimate histories of

Table I. Qualitative Comparison of Techniques Used by Existing Spam Filtering Approaches

Features	P1	P2	P3	P4	P5	P6
Trust computation framework using past history		√		√	√	√
Reputation inference based on recommendations	√	√			√	√
Rate Limiting						√
Presence or context information for real-time analysis						√
Blacklists and whitelists for signature based detection		√	√			√
Community experience	√					√
Feedback control				√		√
Identity Verification		√	√			
Collaborative analysis		√			√	√
Adaptive and real-time usage		√				√
Distributed Solution					√	√
Challenge and Response techniques		√	√			

P1: Golbeck and Hendler [2004] P2: Seigneur et al. [2004]
P3: Wattson [2004] P4: Damiani et al. [2004]
P5: Foukia et al. [2006] P6: Our proposed VoIP spam filtering framework

the call participants. The reputation module infers the reputation (R) of the source domain and then augments the distrust (D) based on inferred reputation. As shown in Figure 6, the final spam behavior (p_c) in filtering is a result of the analysis of the incoming call by applying the principles of distrust and reputation. For the above analysis, the filter formal notation can be summarized as follows:

$D = (C, N_s, N_v)$: Distrust computation based on history of call participants.
R =[$P(X_1|X_2)$, $P(X_1|X_3)$, $P(X_1|X_4)$]: Reputation analysis for the 4-node graph shown before where X_1 is the VSD and X_2, X_3, X_4 are the domain proxies.
$p_c = (D, R)$: A correlation between the trust and reputation analysis. This tells us how the distrust D is updated based on reputation analysis. In this article, we have used a simple linear update based on the extent of deviation in reputation of a domain from its initial reputation.

$F = (p_c, Y)$: F represents a Forward / Reject decision by comparing the final spam level of the call (p_c) with the assumed threshold Y (defined in Axiom 1). F can be a simple Boolean function resulting in *True* (call is spam—filter the call) or *False* (call is not spam—forward the call).

In the above sections, we have discussed an evidence-based filtering technique for Voice over IP. In the next section, we compare our methodology with the existing evidence based techniques.

4.3.4 *Comparison of Evidence-Based Filtering Techniques.* The VoIP spam detection framework discussed previously in this article can be compared with some of the existing evidence based spam filtering approaches used in e-mail. Table I presents a tabular qualitative comparison of the features and techniques supported by different approaches.

5. EXPERIMENTAL RESULTS

VoIP deployment is still at its inception. No VoIP corpus exists for testing a detection mechanism. So, to test our proposed VoIP spam-detection framework,

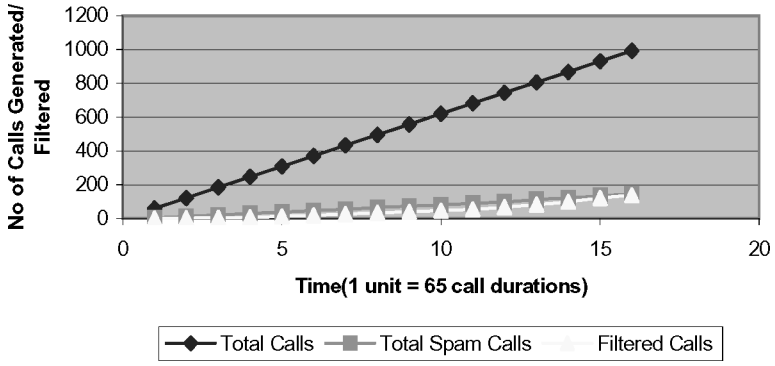

Fig. 7. Comparing the total calls generated, generated spam calls and filtered calls. The VSD continuously tries to catch up with generated spam.

we use randomly generated data for the network setup defined in Figure 1. The end users in the call-generating domains and VSD_{domain} (the enterprise network) are either real SIP IP phones or soft clients compliant with SIP RFC [Rosenberg et al. 2002]. End users outside the VSD_{domain} use randomly generated usernames and IP addresses to form a "*from*" SIP URI. The end users inside the VSD_{domain} (enterprise employees) can receive calls forwarded by the VSD and also generate a call to a randomly selected user outside the VSD_{domain}. The call-generation process uses a Bernoulli distribution. Calls are generated with an average rate of 8 calls/minute. Neither the VSD nor the VSD_{domain} end users have any knowledge of the call-generation process. A random subset of users, hosts, and domains outside the VSD_{domain} are configured to be spam entities before the start of experiments. We ran the experiments with six users inside the VSD_{domain} and 40 users outside the VSD_{domain}. The VSD_{domain} is configured to be a domain with five Class C networks. For a given user inside the VSD_{domain}, Figure 7 compares the number of total calls and spam calls from all users outside the VSD_{domain}, and the number of filtered calls by the VSD.

5.1 VSD Architecture: Collaboration between Different Filtering Techniques

In our architecture, filtering techniques employed at each stage of spam analysis include spam and legitimate signatures (black- and whitelists), trust, and reputation of the calling party. While most current spam filters employ blacklisting as their sole means of stopping junk calls, blacklisting coupled with trust and reputation inference techniques increases the filter accuracy as shown in Figure 8. The figure presents the number of spam calls blocked using the three stages of analysis. It can be observed that the number of spam calls blocked using blacklisting, trust and reputation is approximately 97.16% compared to 4.25% if only blacklisting is implemented.

5.2 Accuracy of the Voice Spam Detector (VSD)

The VSD's accuracy can be estimated by comparing the rate of spam with the rate of filtered calls.

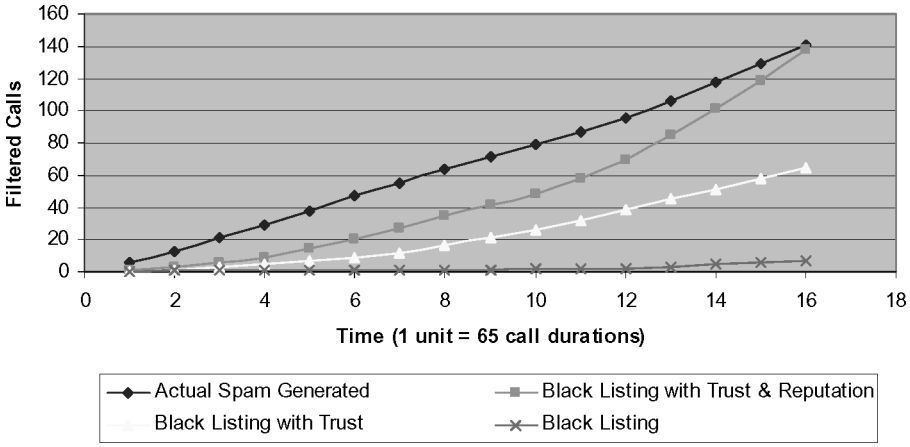

Fig. 8. Spam calls blocked by VSD for different stages of analysis. Filter performance improves significantly when the three stages (blacklisting, trust, and reputation inferences) are used collectively to infer the spam behavior of incoming calls. Data for the plots include spam calls generated by 40 users outside the VSDdomain to a user inside the VSDdomain.

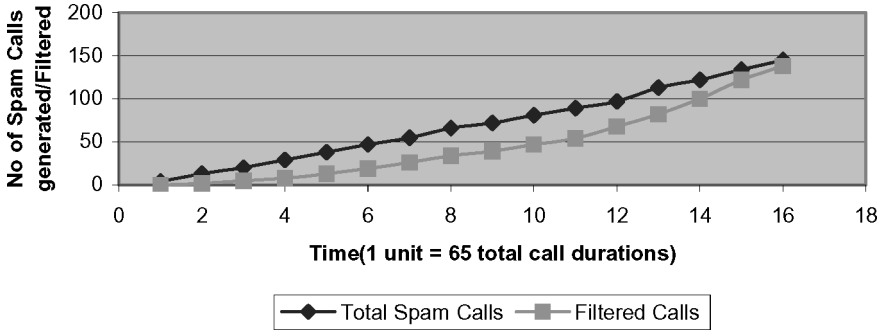

Fig. 9. Spam detection accuracy increases with time. The number of filtered spam calls increase with time as VSD learns the behavior of calling entities. This learned knowledge results in the VSD filtering more and more spam until it catches up with the generated spam.

5.2.1 *Accuracy during Learning Period.* Initially, VSD has no knowledge of spam but learns the callers' behavior using feedback from the end users. This learnt behavior is used for blocking spam calls. The number of spam calls filtered by the VSD increases with time and, ultimately, tries to catch up with the generated spam calls. Figure 9 presents the total number of spam and filtered calls from all the callers to a particular user inside the VSD_{domain}. VSD catches up with spammers during its learning period. After the learning period, VSD has an accuracy of 97.6%, a false positive percentage of 0.4% with 2% of spam calls forwarded to the end user (false negatives). After 16 time units (Figure 9), the filter locks in with the spammers, thus, improving the accuracy of detection.

Similar behavior of filter locking-in with a spammer for a given user inside the VSD_{domain} can be observed in Figure 10.

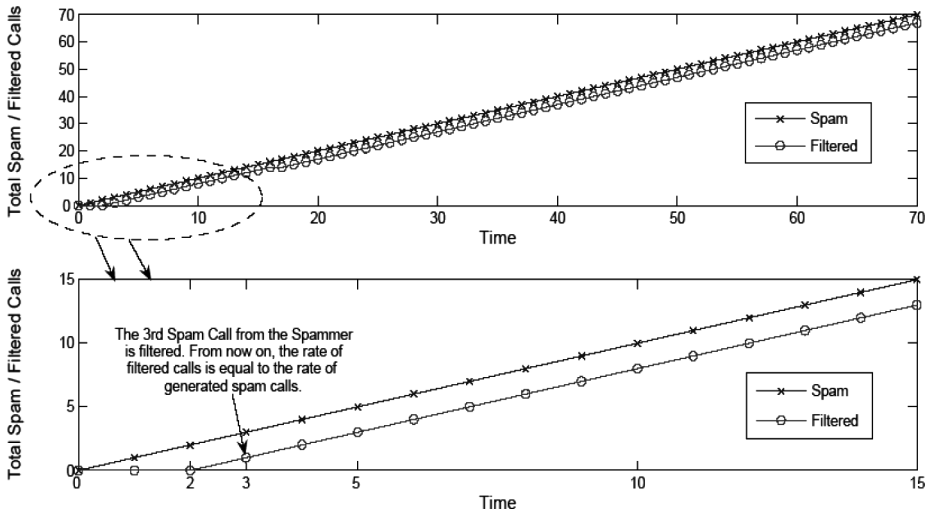

Fig. 10. Spam detection accuracy increases with time. The graph presents the learning period for the user with respect to a particular caller outside the VSDdomain. Initially, spam calls are forwarded, but the VSD learns the caller's spam behavior during the learning period and starts to filter the caller's calls. All spam calls starting with the caller's 3rd spam call are directly filtered.

Figure 10 depicts the VSDs accuracy during the lock-in period for an end user inside the VSD_{domain} from a particular caller. It presents the learning period when the spam calls are generated from that caller. For a caller who is repeatedly spamming the end user, the filtered-calls curve catches up with the spam-calls curve after the 3rd call, that is, the caller's 3rd spam call is automatically filtered. The rate of filtered calls from then on equals the rate of generated spam calls, resulting in minimum false alarms (validating Corollary 1). Next, we study the accuracy of the filter during the lock-in period.

5.2.2 *Accuracy during Lock-In Period.* After the learning period, the number of filtered calls will be close to the number of spam calls. During this time, VSD may filter other calls (creating false positives) or may let the spam calls reach their destination (false negatives).

Figure 11 is a magnified version of Figure 9 during the lock-in period. This graph describes the rate of spam calls versus the rate of filtered calls from all callers to a given user inside the VSD_{domain} during the lock-in period. At any time, the difference between the spam and filtered calls provides the number of false alarms. Initially, the filter starts learning the spam behavior and, therefore, fewer spam calls are filtered, resulting in false negatives. After considerable learning, the rate of filtered calls will almost be equal to the rate of spam calls. It is also possible that the VSD will filter more calls than the actual generated spam calls in that time period resulting in false positives. This can happen in a random setting because some non-spam users can accrue spam behavior by sharing resources (e.g., hosts, domains) with the spammers.

5.2.3 *Improving the Accuracy of VSD.* Filtered calls are fewer than the number of spam calls before lock-in because the VSDs knowledge regarding

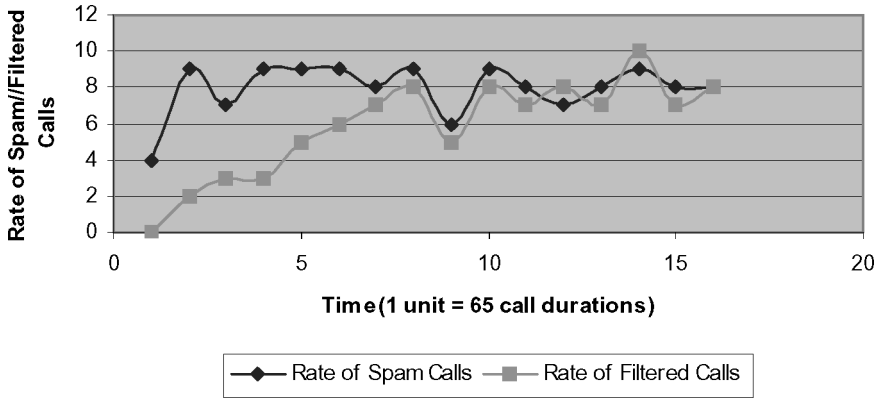

Fig. 11. Small number of false alarms after the learning period. VSD, after learning the caller's spam behavior, filters most of the spam. After the learning period, the calls filtered by VSD are almost same as the spam calls but with very few false alarms (we say that the filter locks in with the spammer).

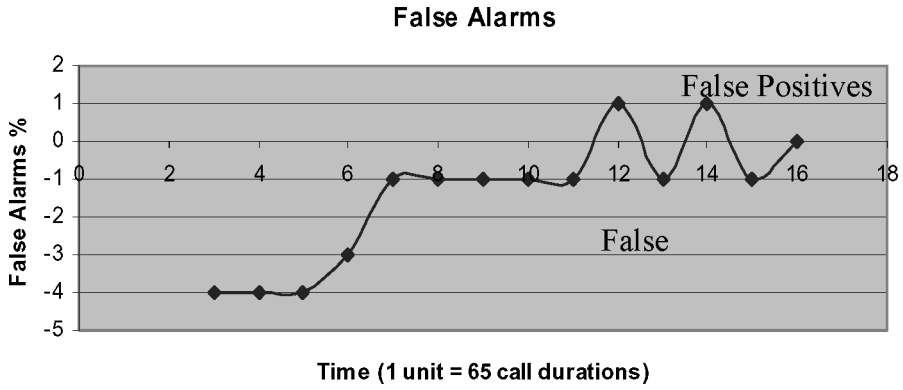

Fig. 12. Small number of false alarms after the learning period. A false negative (spam classified as legitimate) results in ringing the phone and a false positive (legitimate classified as spam) is diverted to the voice mail box. In both cases, the VSD updates the history based on the end-user's feedback. The feedback can either be explicit feedback (when the end-user presses the spam button) or implicit feedback (when the end-user calls back the caller). Hence, user feedback can be used to reduce error and to keep false alarms to a minimum.

the spammers is insufficient. This is the period where false negatives appear. This is represented by the false alarms curve below the x-axis as shown in Figure 12. The curve tends to zero-in when the VSD has the complete history of all the callers so that every spam call can be right away stopped. At times, it can happen that some of the legitimate callers accrue spam behavior by using spam resources (e.g., spam host, spam domain, etc.). Because of this, it is possible that the filter blocks more calls than actual spam calls thereby creating false positives represented by the error curve above the x-axis as shown in Figure 12.

In view of the above scenarios, we believe that in addition to using feedback from the end user regarding false negatives (like a spam button), using feedback about the false positives would prove to be equally effective for spam learning.

This feedback mechanism is similar to e-mail where a filtered email is routed to a junk-mail folder. Instead of directly blocking the filtered calls, the VSD would forward the suspicious call to the user's voice-mail box. The user would have added flexibility of looking at the calls in his voice-mail box and reporting the validity of the calls to the VSD. For example, the user could inform the VSD that a call in the voice-mail box is legitimate. The VSD can then update the legitimate history of that caller. This update can be a linear decrease in distrust or complete spam history reset. In this article, we have used the linear decrease in distrust update procedure. Alternatively, the user could also respond saying that the call is a spam call or could immediately purge the call from the voice-mail box. In this case, the VSD will implicitly understand that the call was indeed spam and that the filter accurately judged the spam behavior of the call. With this kind of end user response, the filter can reduce the number of false positives. As a result, the curve in Figure 12 drops to the actual spam-call curve. This process repeats and, eventually, the filter converges with the Zero line locking-in closely with the spammers.

6. SENSITIVITY ANALYSIS

While considering the accuracy of the filter, it is important to analyze the impact of the configuration parameters. In this article, we analyze the effect of parameters such as spam volume and the network size on the filter's accuracy.

6.1 Spam Volume versus Accuracy

Property 7. More the amount of spam, the easier it is to detect it.

We may receive more spam because of an increasing number of spammers in the call-generating domains outside the $\text{VSD}_{\text{domain}}$ (e.g., telemarketing company with huge employee base) or because the amount of traffic generated by spammers is significantly higher than that of legitimate callers. We have observed that for a given number of calls, the VSD takes less time to learn spamming behavior when the volume of spam calls is high than when the number of spam calls is low. This relationship between spam volume and the VSD's spam-detection capability can be proved using an analytical model. For proving this relation, two different percentages of spam ($x\%$ and $y\%$ of spam processed by VSD such that $x < y$) can be individually substituted in Eq. (1) as shown in A.4 of the Appendix. We further support the analytical model in A.4 of the Appendix by presenting our experimental results for spam-detection capability for varied amounts of spam. To measure this capability, we plot the rate of false negatives for varying amounts of spam.

Figure 13 describes the plot for rate of false negatives versus the amount of spam generated from the calling domain. As the amount of spam increases, the number of false negatives decreases, that is, the filter's spam-detection capability increases. It can be inferred that the detection capability increases with increasing spam encountered, but at the same time the filter shows as much capability with a smaller percentage of spam when it has more time (more calls) for learning. The detection capability based on the experimental results can be compared with that of the analytical model. The detection capability

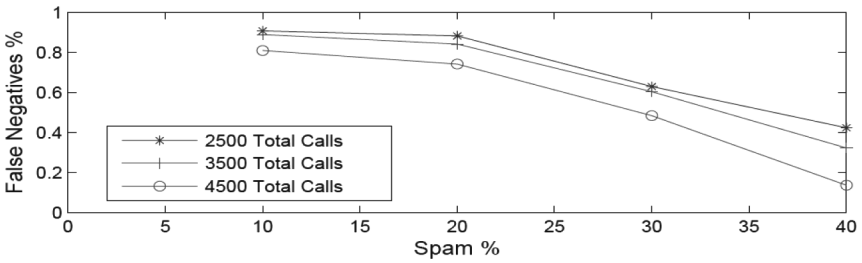

Fig. 13. False negatives decrease when the VSD encounters more spam. The greater the number of spammers among the users generating calls, the higher is the probability that the incoming call is spam. Therefore, the chances that the VSD will filter the call are higher. This larger number of spammers also reduces the chances that the VSD will forward the spam, that is, there are fewer false negatives.

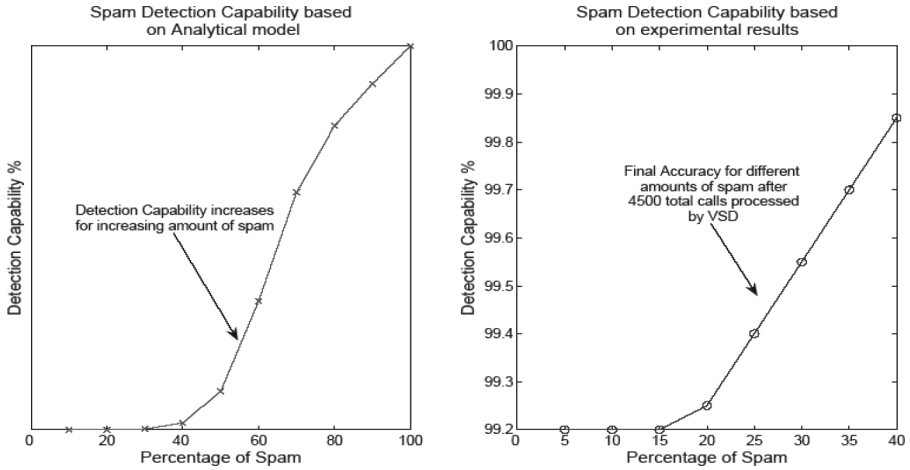

Fig. 14. Experimental Accuracy vs. Analytical Detection Capability with respect to amounts of spam.

based on experimental results is directly derived from the false negative rate shown in Figure 13. The detection capability based on an analytical model can be obtained by substituting different values for $N_{i,s}$ and $N_{i,v}$ (such that the sum of them is always a constant) for a call participant i for $i = 1 \cdot \cdot n$ in Eq. (1).

Figure 14 presents the plot for a comparison between the spam-detection capability based on the analytical model and our experimental results. The spam-detection capability based on experimental results shows similarity with that of the analytical model. However, the increase in the spam-detection capability for larger amounts of spam might result in a few more false positives. This increase can be represented by plotting the false positive rate using RoC curves for two amounts of spam as shown in Figure 15. The figure represents the false positive rate for 20% and 40% of spam in the total traffic processed by VSD. It can be observed that as the amount of spam increases the false positive rate increases. This happens in a random setting because as the number of spammers increase, the percentage of non-spam or legitimate users decreases. Due

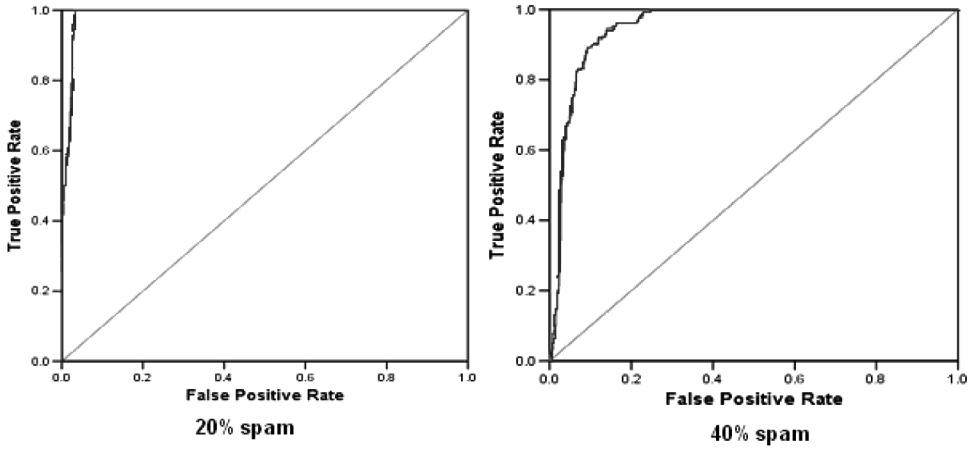

Fig. 15. False positive rate for increasing amounts of spam. The more spammers within the domain, the more likely that valid calls are classified as spam (e.g., because the legitimate callers share the domain or host with the spammers). This sharing results in non-spammers being blocked at the filter resulting in false positives. Therefore, more the spam, the higher are the false positives.

to this, the legitimate users begin to accrue spam behavior by using resources (e.g., host and domain etc.) used by the spammers. This results in the legitimate users getting filtered at the VSD thus resulting in more false positives. Taking into account the feedback about filtered calls as described in Section 5.2.3 can reduce these false positives.

6.2 Network Size versus Accuracy

Another important parameter that can affect the filter's accuracy is network size. The users registering for VoIP services might have dynamic addresses because of the end hosts running the DHCP protocol. Because of this, every time a user connects to the VoIP network, it might have a different Class B or Class C network address (mostly a different Class C). In this event, the number of available IP addresses in the address space greatly affects estimating the spam behavior of the users in the network's VoIP domain. For a smaller number of networks inside a VoIP domain, VoIP users have a more limited set of IP addresses to use for connecting to the network. In this case, the behavior of a spammer having an IP address within this address space can be learned more quickly than is the case when the spammer is in a network with more available IP address space. Catching a spammer who is within a large IP address space becomes more difficult than in the case of a smaller IP address space. We validate this by plotting the number of filtered calls by VSD from the same set of spammers for two sizes of call-generating domains—5 and 10 Class C network domains.

6.2.1 *Network Size versus Blocked or Filtered Calls.* Figure 16 gives the number of spam calls filtered for two different sizes of networks. For the same set of spammers, the time taken by VSD for learning spam from a network size of 10 Class "C" networks is higher when compared to time taken for a network of size 5 Class "C" networks.

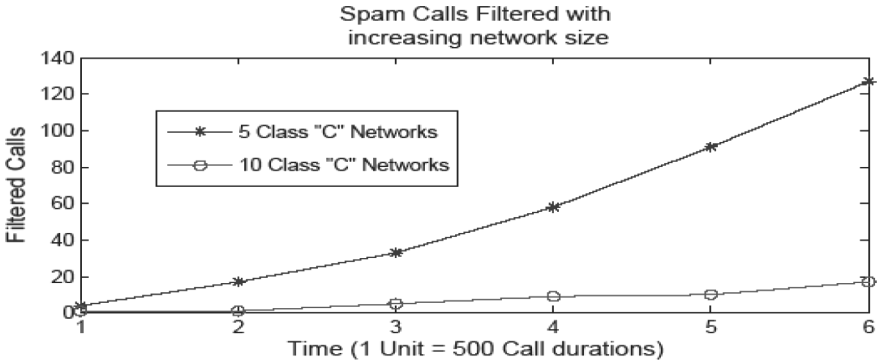

Fig. 16. Spam calls filtered with increasing network size. For a same set of spammers distributed across differently sized networks, VSD is more capable of differentiating spam for smaller-sized networks than larger networks

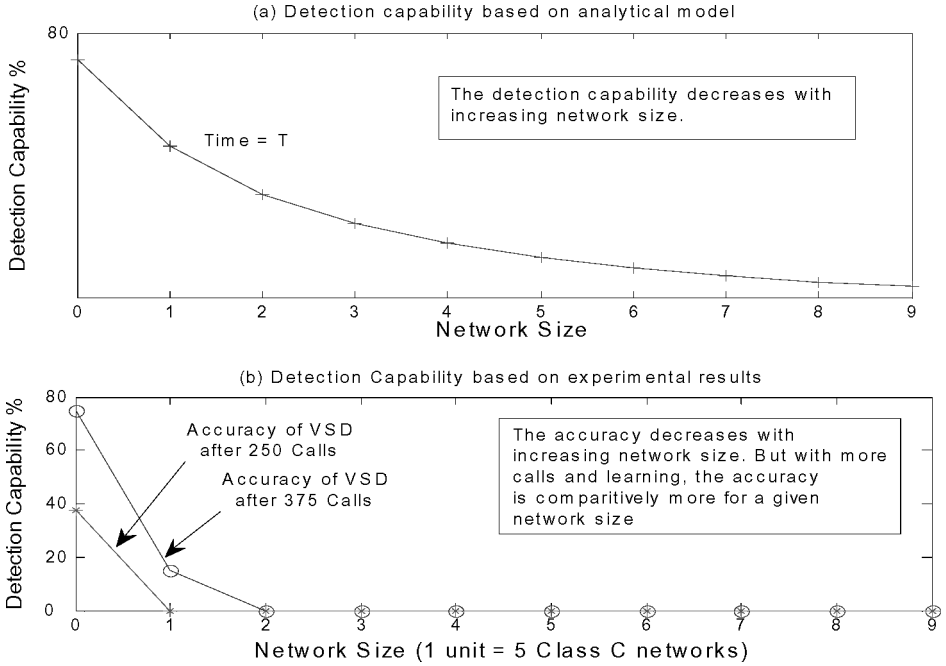

Fig. 17. Blocked spam calls for increasing scalability on call generation. The detection capability based on experimental analysis is similar to that of the analytical model.

The detection capability for network sizes based on the above experimental results and analytical model can be plotted as shown in Figure 17.

Figure 17 shows the relationship between detection capability and network size based on our analytical model and experimental results. The graph for the analytical model is generated by assuming that a fixed number of spammers are distributed over different network sizes. Due to this distribution, the amount of spam varies for each unit of network size, that is, for the same set of spammers,

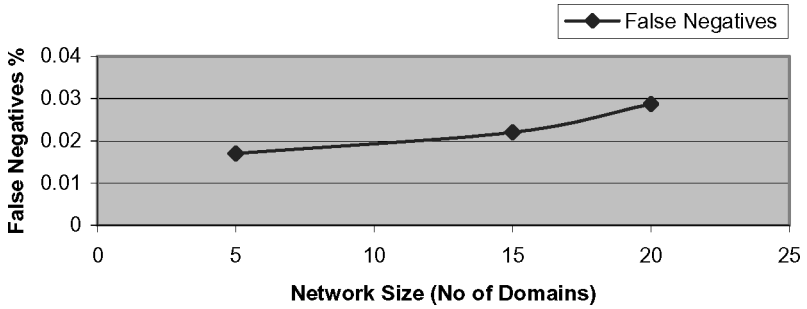

Fig. 18. Increasing false negatives with increasing network size. The false negatives for small-sized networks are fewer than those identified within larger-sized networks due to the smaller number of spam calls at any given time.

the spam generated for each unit of network size (e.g., 1 Class "C" network) in smaller-sized networks is higher when compared to the amount of spam generated for each unit of network size in large-sized networks. Alternatively, the detection capability based on experimental results is directly inferred from the experimental results presented in Figure 16.

From Figure 16 and Figure 17, we note that, for a given time period and number of calls, spam detection is lower for large-sized networks when compared to small-sized networks. However, spam detection is better even for large-sized networks when the filter encounters large number of calls from spammers. In any case, irrespective of the network size, VSD identifies spammers if it receives a sufficient number of calls from those spammers. This can be observed in our daily life as well. It is easy to detect spam from telemarketers from the same company when compared to the telemarketers spread across the country.

6.2.2 *Network Size versus False Alarms.* Here we consider the effect of network size with respect to false alarms.

Figure 18 represents false negative rates for differing network sizes. As the network size increases, VSD takes more time in learning the spam. We need to note that the spam probability of an incoming call depends on the call participants' history. Due to a smaller number of spam calls in each unit-sized networks (e.g., 1 Class C network), it is highly likely that most of the non-spammers have created good will and a legitimate behavior. This will decrease the spam-probability of an incoming call and allow spam to go through VSD. Therefore, with an increasing network size, the allowed spam will increase, that is, number of false negatives increases. We believe we can further reduce the false negatives if we consider the experiences of other's (i.e., neighbors) with the caller. It is highly likely that if a set of users receive spam from a source, the same source will spam other users in the same domain. So, integrating domain-level knowledge can improve the performance of VSD.

6.3 Integrating Domain-Level Knowledge (*We learn from others' experiences*)

It is frequently observed that spammers usually spam more than one user in a domain. This is certainly true for broadcast spammers who spam all the

users in a domain (e.g., multiple employees in the enterprise). For example, a telemarketer would broadcast messages to many users in the domain. Nearly all the recipients of the messages will consider them to be spam. VSD can take advantage of this behavior to identify more spam calls A domain-level database (e.g., stored in a proxy server located in the enterprise's perimeter) can be used for computing the distrust of incoming calls with respect to the callee's community. This domain-level distrust can then be used to either increase or decrease the distrust perceived by the callee (Section 4.3.3). However, in few cases, a callee might want to receive calls that others in the community have categorized as spam. For example, a callee in search of mortgage rates may want to receive broadcast calls about mortgages even though the calls are considered to be spam by many other members in the callee's community. In this case, the customized filter options of the callee (e.g., whitelist) can allow reception of calls of specific interest even when the domain-level spam analysis reports that incoming call is spam. It is also possible that the callee might not have a prior history of calls from such spammers. In this context, if the call is filtered and directed to the callee's voice-mail box, the callee can provide feedback telling the VSD that the call is legitimate. This ensures that future calls from that caller are directly forwarded. On the other hand, if the call is allowed through to the callee and if the callee is interested in taking the call, the callee can provide positive feedback about its legitimacy to the VDS and, thus, increase the caller's trust level relative to the callee.

Property 8. Trusting a caller depends on the social experience of the callee's community.

Upon receiving a call, if the callee doesn't have a prior experience with the caller, the callee can use the social experiences of neighbors to determine if a call should be accepted. These neighbors constitute the callee's community. Some neighbors may have first-hand experience with the caller. This can be taken advantage of for identifying spammers who make spam calls to more than one member in the callee community. In this scenario, the distrust of the incoming call with respect to all the members in the callee's community is given by

$$D^d = f(\{N^d_{1,s}, N^d_{1,v}\}, \{N^d_{2,s}, N^d_{2,v}\}, \{N^d_{3,s}, N^d_{3,v}\}, \ldots, \{N^d_{n,s}, N^d_{n,v}\}, N_S, N_V),$$

where $N^d_{i,s}$ and $N^d_{i,v}$ represents the spaminess and legitimateness of call participant i with respect to all the users inside the $\text{VSD}_{\text{domain}}$, that is, callee's community (e.g., all enterprise employees). Simply, for m number of callee community members, $N^d_{i,s} = \sum_{k=1}^{m} N_{i,s}$where $N_{i,s}$ is the spaminess of the call participant i with respect to the community member k. Similarly, for m number of callee community members, $N^d_{i,v} = \sum_{k=1}^{m} N_{i,v}$ where $N_{i,v}$ is the legitimateness of the call participant i with respect to the community member k. However, $N_{i,s}$ and $N_{i,v}$ may take different values when considered for different callees (community members). In order to have a lighter notation, we avoided indexing $N_{i,s}$ and $N_{i,v}$ by callee in the rest of the article, but $N^d_{i,s}$ and $N^d_{i,v}$ actually are $\sum_{k=1}^{m} N^k_{i,s}$and $\sum_{k=1}^{m} N^k_{i,v}$ respectively.

Note. The above scenario assumes that each member of the callee's community is of equal importance to the callee. It is quite possible that the callee himself has different trust relationships with each community member. In this context, the callee may find it more useful to obtain recommendations from all the community members. The callee can then weigh the recommendations based on the trust relationships the callee holds with each member who has responded to the request. The community feedback to callee "a" about caller "b" can thus be represented by

$$T(a,b) = \Delta(T(u_1,b), T(u_2,b), T(u_3,b), \ldots, T(u_m,b)),$$

where $T(u_k,b)$ for $k = 1 \cdot m$ represents the trust of member u_k towards caller"b"

The function Δ can be a simple weighted function as below.

$$T(a,b) = W_1T(u_1,b) + W_2T(u_2,b) + W_3T(u_3,b) + \cdots + W_mT(u_m,b),$$

such that $W_1 + W_2 + W_3 + \ldots W_m = 1$

The weight constants $W_1, W_2, \ldots . W_m$represent the trust for the callee towards each of the community members.

Property 9. The larger the community, the smaller the impact of an individual.

The callee requests the neighbors about the trust values they associate with a caller when the callee lacks first hand information about the caller. If the set of callee's neighbors who respond is large, the individual significance of each recommendation decreases. On the other hand, if the set of neighbors who respond to the request is small, the weighted response of each neighbor has a more influential effect on the callee's decisions.

6.3.1 *Increase in Performance of VSD by Integrating Domain-Level Knowledge.* To measure the increase in performance of VSD by integrating domain level knowledge, we ran the experiment for 1500 calls with six users in the $\text{VSD}_{\text{domain}}$. VSD logs the spam and legitimate calls to all the callees inside the $\text{VSD}_{\text{domain}}$. In addition to computing the callee specific distrust value, the VSD also computes the call's distrust with respect to the callee's community. The VSD then uses both the distrust values for inferring the spam behavior of the call.

Figure 19 represents the increase in the accuracy during the learning period by integrating domain knowledge for a user inside the $\text{VSD}_{\text{domain}}$. It can be seen that by integrating the domain level spam information for distrust computation, the increase in performance of the VSD is as high as 100% (Figure 19(a)). In addition, as shown in Figure 19(b), a marked improvement is observed in the false negative rate by integrating domain knowledge.

7. CONCLUSION

The problem of spam in VoIP networks has to be solved in real time, unlike e-mail systems. Many of the techniques devised for e-mail spam detection rely upon content analysis. But, in the case of VoIP, it is too late to analyze the media (content) as the parties are already in communication. So, we need to stop the

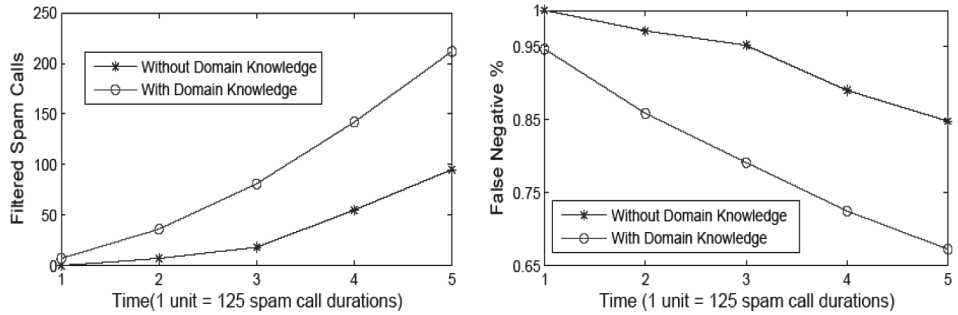

Fig. 19. Increase in filter accuracy by integrating domain level information. The broadcasting mechanism followed by many spammers can be taken advantage of to block spam across the domain. Integrating this knowledge helps in identifying true spam and, therefore, in reducing false negatives.

spam calls before the telephone rings. To meet this goal, we developed a five-stage process for determining whether an incoming call is spam. These stages include multivariable Bayesian analysis for computing and updating trust, and Bayesian Networks for inferring reputation. The results from each stage are fed back for collaboration between processes. In addition, we presented formalism on how we computed and updated trust and reputation for use in VoIP spam filtering. This formalism is based on human intuitive behavior in responding to a received call and updating the trust and reputation based on one's experience. We have also verified our results using an experimental setup consisting of VoIP soft clients, real IP phones, and a commercial-grade SIP proxy server. Due to the unavailability of real VoIP calling patterns, we simulated random traffic consisting of a set of users. Few of those users are preconfigured to be spammers. Calls are generated among the users through the VSD and the spam filter (VSD) analyzes each call going through it. From our observation of the logs, while the spam filter directly filters the 2nd call from a spammer calling from the same end host and domain, it needs a maximum of 3 spam calls to learn the behavior when the spammer changes his host and domain associations' that is, when the spammer calls from a different host and domain. In the end, we presented a detailed sensitivity analysis demonstrating the filter's accuracy with respect to important configuration parameters such as spam volume and network size. Finally, to reduce the false alarms, we have presented a domain-based feedback mechanism using knowledge about a caller drawn from the callee's community. Using our spam detection framework, we achieved an average performance of 96% in filtering VoIP spam calls. However, our proposed solution is not a single bullet proof solution and can be used with other identity verification algorithms for more optimum performance.

APPENDIX

A.1 Terminology

—A caller/calling party is a person/entity generating a call.

—A callee/called party is a person/entity receiving a call.

—A call participant can be user, host, domain, proxy in the path, etc.

—Spaminess: Amount of spam behavior or the associated spam history. This is given by the number of past spam calls.

—Legitimateness: Amount of valid behavior or the associated non-spam history. This is given by number of past legitimate calls.

—n: Number of call participants in an incoming call. $n \in N$ (set of natural numbers).

—i: Refers to the ith call participant.

—N_S: Total number of spam calls processed by VSD. $N_S \in N$.

—N_V: Total number of legitimate calls processed by VSD. $N_V \in N$.

—$N_{i,s}$: Spaminess of a call participant i . Spaminess refers to the number of spam calls of the participant. $N_{i,s} \in N$.

—$N_{i,v}$: Legitimateness of a call participant i. Legitimateness refers to the number of legitimate calls of the participant. $N_{i,v} \in N$.

—C_i: A call set of participant i. The call set C_i for participant i is given by $\{N_{i,s}, N_{i,v}\}$.

—C: Set of call sets of all participants. $C = \{C_1, C_2, \ldots, C_n\}$.

—$N_{i,B}$: Total calls from a participant i. $N_{i,B}$ is the summation of spaminess and legitimateness of the participant, that is, $N_{i,B} = N_{i,s} + N_{i,v}$. $N_{i,B} \in N$.

—m: Number of callee community members. $m \in N$

—u_k: kth member in the callee's community such that $1 \leq k \leq m$ and $k \in N$

—$N^d_{i,s}$: Spaminess (total spam calls) of a call participant i observed by callee's domain. The superscript "d" represents the domain-wide scope of i. $N^d_{i,s} \in N$.

—$N^d_{i,v}$: Legitimateness (total valid calls) of a call participant i observed by callee's domain. $N^d_{i,v} \in N$.

—T_i: Trust level of a call participant i. $T_i \in [0\ 1]$.

—$T_{i,t}$: Trust level of a call participant i at time$t$. $T_{i,t} \in [0\ 1]$.

—T: Trust of an incoming call. $T \in [0\ 1]$.

—D: The distrust of an incoming call . $D \in [0\ 1]$.

—D_i: The distrust level of a call participant i. $D_i \in [0\ 1]$.

—D^d: The distrust of an incoming call observed by callee's community. $D^d \in [0\ 1]$.

—N_{CF}: Number of spam or valid calls that can move current distrust level D_C to a final distrust level D_F . $N_{CF} \in N$.

—R: Reputation of the calling party. $R \in [0\ 1]$.

—p_c: Spam probability of an incoming call based on trust and reputation. $p_c \in [0\ 1]$.

—Y: The callee's tolerance (threshold) towards the spam behavior of the incoming call. $Y \in [0\ 1]$.

—F: Forward or Filter decision. This decision F can be simple Boolean function resulting in *True* (filter the call) or *False* (forward the call). $F \in$ {True, False}.

A.2 Probabilistic Model for Computing the Probability of the Call to be Spam using Naïve Bayesian

Given an instance M for the incoming call and values of $n_1, n_2, n_3, \ldots, n_n$ to the feature variables, the probability of the instance M to have a classification CL can be computed using Bayes theorem.

$$P(CL = cl_k|M = n) = \frac{P(CL = cl_k)P(M = n|CL = cl_k)}{P(M = n)}.$$

Using a Naïve Bayesian classifier [Good 1965; Sahami et al. 1998] that assumes that every feature is conditionally independent of all other features, we have

$$P(M = n|CL = cl_k) = \prod_{i=1}^{n} P(M_i = n_i|CL = cl_k)$$

Therefore,

$$P(CL = cl_k|M = n) = \frac{P(CL = cl_k)\prod_{i=1}^{n} P(M_i = n_i|CL = cl_k)}{P(M = n)}.$$

From Bayes theorem, we also have

$$P(M = n) = \sum_{k} P(M = n|CL = cl_k)P(CL = cl_k)$$

for k different classifications of the instance M.

Therefore,

$$P(CL = cl_k|M = n) = \frac{P(CL = cl_k)\prod_{i=1}^{n} P(M_i = n_i|CL = cl_k)}{\sum_k P(M = n|CL = cl_k)P(CL = cl_k)}$$

We use the above Naïve Bayesian Probabilistic model for inferring spam behavior of incoming calls. The model is used for classifying the incoming call instance into two different classifications of call being spam or the call being legitimate.

For this, we define the features of call instance to be the calling user, calling host and call-generating domain. Each of the above call features is independent to others.

From the above equation, the probability of the call being classified as spam is given by

$$P(CL = spam|M = n) = \frac{P(CL = spam)\prod_{i=1}^{n} P(M_i = n_i|CL = spam)}{\sum_{c_k \in \{spam, legitimate\}} P(M = n|CL = cl_k)P(CL = cl_k)},$$

$$P(CL = spam|M = n) = \frac{P(CL = spam)\prod_{i=1}^{n} P(M_i = n_i|CL = spam)}{P(CL = spam)\prod_{i=1}^{n} P(M_i = n_i|CL = spam) + P(CL = legitimate)\prod_{i=1}^{n} P(M_i = n_i|CL = legitimate)} \quad (13)$$

Each term in the above equation can be computed by logging the feedback from all the called parties for all the forwarded calls to them. In the above

equation, $P(CL = spam)$ represents the probability of spam calls processed by the VSD and it can be estimated from the logged history, that is,

$$P(CL = spam) = \frac{Total\ spam\ calls}{Total\ spam\ calls + Total\ legitimate\ calls} = \frac{N_S}{N_S + N_V}$$

where N_S and N_V represent the total number of spam and legitimate calls seen before.
Similarly,

$$P(CL = legitimate) = \frac{Total\ legitimate\ calls}{Total\ spam\ calls + Total\ legitimate\ calls} = \frac{N_V}{N_S + N_V}.$$

However, for negligible N_S compared to N_V (i.e., when the number of observed previous spam calls is negligible compared to legitimate calls), we have P($CL = legitimate$)$\rightarrow$ 1. In this case, every incoming call would be initialized to a high legitimate probability when substituted in Eq. (13). Therefore, in this context, few spam calls go unnoticed resulting in false negatives.

Similarly, for a negligible N_V compared to N_S, (i.e., when the number of observed previous legitimate calls is negligible compared to spam calls), we have P($CL = spam$)$\rightarrow$ 1. In this case, every incoming call would be initialized to a high spam probability when substituted in Eq. (13). In this context, legitimate calls get filtered at the spam filter.

Therefore, to reduce the influence of above two terms in call classification, we compute those terms based only on the histories of the participating entities of the call such as the calling user, calling host and call-generating domain, that is,

$$P(CL = spam) = \frac{Total\ spam\ calls\ from\ call\ participants}{\begin{matrix}Total\ spam\ calls \\ from\ call\ participants\end{matrix} + \begin{matrix}Total\ legitimate\ calls\ from \\ call\ participants\end{matrix}}$$

and

$$P(CL = legitimate) = \frac{Total\ \ legitimate\ calls\ from\ \ call\ participants}{\begin{matrix}Total\ spam\ calls \\ from\ \ call\ participants\end{matrix} + \begin{matrix}Total\ legitimate\ \ calls\ from \\ call\ participants\end{matrix}}$$

that is,

$$P(CL = spam) = \frac{\sum_{i=1}^{n} N_{i,s}}{\sum_{i=1}^{n} N_{i,s} + N_{i,v}}$$

and

$$P(CL = legitimate) = \frac{\sum_{i=1}^{n} N_{i,v}}{\sum_{i=1}^{n} N_{i,s} + N_{i,v}}$$

Also,

$$\begin{matrix}\text{Probability for a call participant } n_i \text{ to be spam} \\ = P(M_i = n_i | CL = spam) = \frac{N_{i,s}}{N_{i,s}+N_{i,v}}\end{matrix}$$

And probability for a call participant n_i to be legitimate
$$= P(M_i = n_i | C = legitimate) = \frac{N_{i,v}}{N_{i,s}+N_{i,v}}$$

Therefore,

$$P(CL = spam|M = n) = \frac{\left(\frac{\sum_{i=1}^{n} N_{i,s}}{\sum_{i=1}^{n}(N_{i,s}+N_{i,v})}\right)\prod_{i=1}^{n}\frac{N_{i,s}}{N_{i,s}+N_{i,v}}}{\left(\frac{\sum_{i=1}^{n} N_{i,s}}{\sum_{i=1}^{n}(N_{i,s}+N_{i,v})}\right)\prod_{i=1}^{n}\frac{N_{i,s}}{N_{i,s}+N_{i,v}} + \left(\frac{\sum_{i=1}^{n} N_{i,v}}{\sum_{i=1}^{n}(N_{i,s}+N_{i,v})}\right)\prod_{i=1}^{n}\frac{N_{i,v}}{N_{i,s}+N_{i,v}}}.$$

A.3. Bayesian Network Probabilistic Model for Deriving Reputation Information

A Bayesian network is a graphical model for showing probabilistic relationships among a set of variables in a Directed Acyclic Graph (DAG). A Directed Acylic Graph contains a set of nodes and directed links between them where each node is a variable and the links connecting two nodes in a DAG are the dependencies existing between those two variables. For an n node graph represented by n random variables $V_1, V_2, V_3, \ldots, V_n$, the probability distribution function would be equal to

$$P(V_1, V_2, \ldots, V_n) = P(V_1)P(V_2|V_1)\cdots P(V_n|V_1 \cdot \cdot V_{n-1}).$$

However, the values of a node are conditioned only on its parents
Therefore,

$$P(V_1, V_2, \ldots, V_n) = \prod_{i=1}^{n} P(V_i/parents(V_i)),$$

where every joint probability$P(V_i|V_j)$ can be expanded into sum of two joint probabilities to include the parent of a variable and can be shown as

$$P(V_i|V_j) = P(V_i, V_k|V_j) + P(V_i, \sim V_k|V_j)$$

for parent node V_k of node V_i and $i, j, k \in 1 \cdot \cdot n$

A conditionalized version of chain rule is given by

$$P(V_i, V_k|V_j) = P(V_i|V_k, V_j)P(V_k|V_j) \text{ for } i, j, k \in 1 \cdot \cdot n$$

The above chain rule can be used for deducing the posterior probability of a random variable for observed evidence. If $P(V_i)$ for $i = 1 \cdot \cdot n$ represents the previous subjective belief of a variable V_i, then $P(V_i|V_j)$ represents the posterior probability of V_i for an observed evidence at V_j that is, the observed evidence at node represented by variable V_j can be propagated throughout the graph by computing $P(V_i|V_j)$ for $i \neq j$ and $i, j \in 1 \cdot \cdot n$.

For a given directed graph represented in Figure 20, assume that the nodes represent the proxy nodes of domains that help in routing the call. Calls are generated from different users in different domains to an end user in the domain X_1 for which VSD acts as a spam filtering device. Assume $P(X_i)$ represents the reputation (initial subjective belief) of domain X_i for $i = 2 \cdot \cdot 4$ with respect to the VSD. Each value $P(X_i)$ for $i = 2 \cdot \cdot 4$ ranges between 0 and 1.

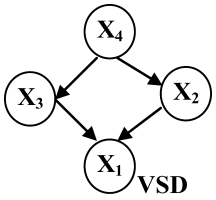

Fig. 20. Network topology graph of domain proxies.

Assumptions. The random variables associated with these proxy nodes are independent. The reputation of a given domain X_i for $i = 1 \cdot \cdot 4$ does not depend on the reputation of other domain X_j for $j = 1 \cdot \cdot 4$ and $i \neq j$.

For observed evidence that a call from domain X_4 to X_1 is spam (when the end user gives a feedback to the VSD that the received call is spam), the updated reputation of domain X_4 with respect to VSD can be inferred by $P(X_1|X_4)$. Using the conditionalized version of chain rule,

$$\begin{aligned} P(X_1|X_4) &= P(X_1, X_2|X_4) + P(X_1, \sim X_2|X_4) \\ &= P(X_1|X_2)P(X_2|X_4) + P(X_1| \sim X_2)P(\sim X_2|X_4), \end{aligned}$$

where

$$\begin{aligned} P(X_1|X_2) &= P(X_1, X_3|X_2) + P(X_1, \sim X_3|X_2) = P(X_1|X_2, X_3)P(X_3) \\ &\quad + P(X_1|X_2 \sim X_3)P(\sim X_3) \\ P(X_1| \sim X_2) &= P(X_1, X_3| \sim X_2) + P(X_1, \sim X_3| \sim X_2) \\ &= P(X_1| \sim X_2, X_3)P(X_3) + P(X_1| \sim X_2, \sim X_3)P(\sim X_3) \text{ and} \\ P(X_3) &= P(X_3, X_4) + P(X_3, \sim X_4) = P(X_3|X_4)P(X_4) \\ &\quad + P(X_3| \sim X_4)P(\sim X_4). \end{aligned}$$

Solving the above equations gives the updated reputation $P(X_1|X_4)$ of domain proxy X_4.

A.4. Analytical Model for Deriving the Relation between the Amount of Spam and Spam Detection Capability

To determine the relationship between the spam detection capability for varying amounts of spam, assume that the filter processes $x\%$ and $y\%$ of spam in a given time t. Assume that in both the cases the filter processes a total of "t" calls. Assume also that $x < y$.

Now, for a call with n call participants, the distrust D is given by

$$D = \frac{\left(\frac{\sum_{i=1}^{n} N_{i,s}}{\sum_{i=1}^{n}(N_{i,s}+N_{i,v})}\right)\prod_{i=1}^{n}\left(\frac{N_{i,s}}{N_{i,s}+N_{i,s}}\right)}{\left(\frac{\sum_{i=1}^{n} N_{i,s}}{\sum_{i=1}^{n}(N_{i,s}+N_{i,v})}\right)\prod_{i=1}^{n}\left(\frac{N_{i,s}}{N_{i,s}+N_{i,s}}\right) + \left(\frac{\sum_{i=1}^{n} N_{i,v}}{\sum_{i=1}^{n}(N_{i,s}+N_{i,v})}\right)\prod_{i=1}^{n}\left(\frac{N_{i,v}}{N_{i,s}+N_{i,s}}\right)}$$

from Eq. (1)

For both the amounts of spam, the factors

$$\prod_{i=1}^{n} \frac{N_{i,s}}{N_{i,s} + N_{i,v}}$$

and

$$\prod_{i=1}^{n} \frac{N_{i,v}}{N_{i,s} + N_{i,v}}$$

are constant as these two factors depend upon the spaminess and legitimateness of individual call participants rather than the number of spam and legitimate calls processed by the spam filter.

Therefore, for $x\%$ of spam, distrust D is given by $\frac{x.a}{x.a+(t-x)b}$, where

$$\left(\frac{\sum_{i=1}^{n} N_{i,s}}{\sum_{i=1}^{n} (N_{i,s} + N_{i,v})}\right) = x$$

(i.e, $x\%$ of calls are spam calls among the total calls), and a, b are assumed to

$$\prod_{i=1}^{n} \frac{N_{i,s}}{N_{i,s} + N_{i,v}}$$

and

$$\prod_{i=1}^{n} \frac{N_{i,v}}{N_{i,s} + N_{i,v}}$$

respectively. Similarly for $y\%$ of spam, the distrust is given by $\frac{y.a}{y.a+(t-y)b}$ since

$$\left(\frac{\sum_{i=1}^{n} N_{i,s}}{\sum_{i=1}^{n} (N_{i,s} + N_{i,v})}\right) = \mathrm{y}$$

(i.e., $y\%$ of calls are spam calls), and a, b are same as above

Now,

$$\begin{aligned}
&\frac{x.a}{x.a + (t-x)b} < \frac{y.a}{y.a + (t-y)b} \\
&\Rightarrow xya + xb(t-y) < xya + yb(t-x) \\
&\Rightarrow x(t-y) < y(t-x) \\
&\Rightarrow x < y
\end{aligned}$$

Conversely, if $x < y$, then

$$\frac{x.a}{x.a + (t-x)b} < \frac{y.a}{y.a + (t-y)b},$$

that is, the distrust for a call for less previously observed spam ($x\%$) by the VSD is less than distrust for the call for more previously observed spam ($y\%$). It is easy to filter out spam calls with more spam behavior or distrust than the one's with less spam behavior. As $x < y$, therefore, it can be understood that more the amount of spam, easier to detect it.

REFERENCES

BIEVER, C. 2004. Move over spam, make way for "spit". http://www.newscientist.com/article.ns?id=dn6445

BOYKIN, P.O. AND ROYCHOWDHURY, V. 2004. Personal Email networks: An effective Anti-spam tool. *Preprint, http://www.arxiv.org/abs/cond-mat/0402143*

CAHILL, V., SHAND, B., GRAY, E., DIMMOCK, N., TWIGG, A., BACON, J., ENGLISH, C., WAGEALLA, W., TERZIS, S., NOXON, P., BRYCE, C., SERUGENDO, G.M., SEIGNEURL, J. M., CARBONE, M., KRUKOW, K., JENSON, C., CHEN, Y., AND NIELSEN, M. 2003. Using trust for secure collaboration in uncertain environments. *IEEE Pervas. Comput. 2*, 3, 52–61.

COHEN, W. W. 1996. Learning rules that classify e-mail. In *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*.

DAMIANI, E., VIMERCATI, S. D. C., PARABOSCHI, S., AND SAMARATI, P. 2004. P2P-Based collaborative spam detection and filtering. In *Proceedings of 4th IEEE Conference on Peer-to-Peer Computing (P2P'04)* (Zurich, Switzerland). IEEE Computer Society Press, Los Alamitos, CA.

DANTU, R. AND KOLAN, P. 2004. Preventing Voice Spamming. In *Proceedings of the IEEE GlobeComm Workshop on VoIP Security*. IEEE Computer Society Press, Los Alamitos, CA.

DANTU, R. AND KOLAN P. 2005. Detecting spam in VoIP networks. In *Proceedings of USENIX, SRUTI(Steps for Reducing Unwanted Traffic on the Internet) Workshop.*

DANTU, R., CANGUSSU, J., AND YELIMELI, A. 2004b. Dynamic control of worm propagation. In *Proceedings of the IEEE International Conference on Information Technology (ITCC)*.

EVETT, D. 2006. Spam Statistics 2006. http://spam-filter-review.toptenreviews.com/spam-statistics.html.

FOUKIA, N., ZHOU, L., AND NEUMAN, C. 2006. Multilateral decisions for collaborative defense against unsolicited bulk e-mail. In *Proceedings of the International Conference on Trust Management*.

GOECKS, J. AND MYNATT E. D. 2002. Enabling privacy management in ubiquitous computing environments through trust and reputation systems. In *Proceedings of the Workshop on Privacy in Digital Environments: Empowering Users. Proceedings of CSCW*.

GOLBECK, J. AND HENDLER, J. 2004. Reputation network analysis for email filtering. In *Proceedings of the IEEE conference on Email and Anti Spam*. IEEE Computer Society Press, Los Alamitos, CA.

GOOD, I. J. 1965. *The estimation of probabilities: An essay on modern Bayesian methods*. M.I.T Press, Cambridge, MA.

HEPBURN, M. AND WRIGHT, D. 2003. Execution contexts for determining trust in a higher-order π calculus. School of Computing, University of Tasmania Technical Report.

JØSANG, A., ISMAIL, R., AND BOYD, C. 2006. A survey of trust and reputation systems for online service provision. *Decision Support Systems*.

KRUKOW, K. AND NIELSEN, M. 2006. From simulations to theorems: A position paper on research in the field of computational trust. In *Proceedings of Formal Aspects in Security and Trust*.

LACY, S. 2006. Is your VoIP phone vulnerable? http://www.businessweek.com/technology/content/jun2006/tc20060613_799282.htm

LANCASTER, K. 2003. Resilient packet ring: Enabling VoIP delivery. *Internet Telephony*.

LEI, H. AND SHOJA, G. C. 2005. A distributed trust model for e-commerce applications. *IEEE International Conference on e-Technology, e-Commerce and e-Service*.

MACINTOSH, R. AND VINOKUROV, D. 2005. Detection and mitigation of spam in IP telephony networks using signaling protocol analysis. In *Proceedings of the IEEE Symposium on Advances in Wired and Wireless Communication*. IEEE Computer Society Press, Los Alamitos, CA. 49–52.

MARSH, S. 1994. Formalizing trust as a computational concept. Ph.D. dissertation. University of Stirling.

MUI, L., MOHTASHEMI, M., AND HALBERSTADT, A. 2002. A computational model of trust and reputation. In *Proceedings of the 35th Hawaii International Conference on System Science.* 280–287.

NICCOLINI, S., TARTARELLI, S., STIEMERLING, M., AND SRIVASTAVA, S. 2006. SIP extensions for SPIT identification. IETF SIP draft, draft-niccolini-sipping-feedback-spit-02.

ONO, K. AND SCHULZRINNE, H. 2005. Trust path discovery. *IETF Internet Draft*.

ORBAEK, P. AND PALSBERG, J. 1997. Trust in the λ calculus. *Funct. Prog. 7*, 6, 557–591.

PALLA, S. AND DANTU, R. 2006. Detecting Phishing in Emails. *Spam Conference, MIT*.

RAGO, S. 2006. VoIP spells equipment oppurtunities now. Networking and Optical Communications—Q3 Topical Report, Isuppli.

RAHMAN, A. A. AND HAILES, S. 1998. A distributed trust model. In *Proceedings of New Security Paradigms Workshop*, ACM Press, New York, 48–60.

RAY, I. AND CHAKRABORTY, S. 2004. A vector model of trust for developing trustworthy systems. In *Proceedings of 9th European Symposium on Research in Computer Security (ESORICS'04)*, (Sophia Antipolis, France).

REBAHI, Y. AND SISALEM, D. 2005. SIP service providers and the spam problem. In *Proceedings of Voice over IP Secuity Workshop* (Washington, DC).

RIGOUTSOS, I. AND HUYNH, T. 2004. Chung-Kwei: A pattern discovery based system for the automatic identification of unsolicited e-mail messages. In *Proceedings of the 1st Conference on E-mail and Anti-Spam*.

ROSENBERG, J., SHULZRINNE, H., CAMERILLO, G., JOHNSTON, A., PETERSON, J., SPARKS, R., HANDLEY, M., AND SCHOOLER, E. 2002. Session Initiation Protocol. RFC 3261

ROSENBERG, J., JENNINGS, C., AND PETERSON, J. 2006. The session initiation protocol (SIP) and spam. Spam Draft - *draft-ietf-sipping-spam-02.txt*

SABATER, J. AND SIERRA, C. 2005. Review on computational trust and reputation models. *Artifi. Intell. Rev. 24*, 33–60.

SAHAMI, M., DUMAIS, S., HECKERMAN, D., AND HORVITZ, E. 1998. A Bayesian approach to filtering junk e-mail. Learning for Text Categorization—*Papers from the AAAI Workshop, pp. 55–62*, Madison, WI. AAAI Technical Report WS-98-05.

SAKKIS, G., ANDROUTSOPOULOS, I., PALIOURAS, G., KARKALETSIS, V., SPYROPOULOS, C. D., AND STAMATOPOULOS, P. 2003. A memory-based approach to anti-spam filtering for mailing lists. *Inf. Retrieval*.

SEIGNEUR, J. M., DIMMOCK, N., BRYCE, C., AND JENSEN, C. D. 2004. Combating spam with TEA (Trustworthy email addresses). In *Proceedings of the 2nd Annual Conference on Privacy, Security and Trust (PST'04)* (Fredericton, New Brunswick, Canada). 47–58.

SHIN, D. AND SHIM, C. 2005. Voice spam control with gray leveling. In *Proceedings of 2nd VoIP Security Workshop* (Washington, DC).

SOONTHORNPHISAJ, N., CHAIKULSERIWAT, K., AND TANG-ON, P. 2002. Anti-spam filtering: A centroid based classification approach. In *IEEE Proceedings ICSP*. IEEE Computer Society Press, Los Alamitos, CA.

WANG, Y. AND VASSILEVA, J. 2003a. Bayesian network-based trust model. In *Proceedings of IEEE/WIC International Conference on Web Intelligence (WI 2003)*. IEEE Computer Society Press, Los Alamitos, CA.

WANG, Y. AND VASSILEVA, J. 2003b. Bayesian network-based trust model in peer-to-peer networks. In *Proceedings of the Workshop on "Deception, Fraud and Trust in Agent Societies" at the Autonomous Agents and Multi Agent Systems (AAMAS-03)* (Melbourne, Australia).

WATTSON, B. 2004. Beyond identity: Addressing problems that persist in an electronic mail system with reliable sender identification. In *Proceedings of the 1st Conference on Email and Anti-Spam (CEAS)*.

YU, B. AND SINGH, M. P. 2002. An evidential model of distributed reputation management. In *Proceedings of 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Vol. 1, ACM, New York, 294–301.

YU, B. AND SINGH, M. P. 2001. Towards a probabilistic model of distributed reputation management. In *Proceedings of 4th Workshop on Deception, Fraud and Trust in Agent Societies* (Montreal, Canada).

ZACHARIA, G. AND MAES, P. 2000. Trust management through reputation mechanisms. *Appl. Artifi. Intell. 14*, 9, 881–908.

ZACHARIA,G., MOUKAS, A. AND MAES, P. 1999. Collaborative reputation mechanisms in electronic marketplaces. In *Proceedings of 32nd Hawaii International Conference on System Sciences*.

ZIMMERMAN, P. R. 1995. The Official PGP User's Guide. MIT Press, Cambridge, MA.

Received June 2006; revised November 2006; accepted November 2006