LENS: Leveraging Social Networking and Trust to Prevent Spam Transmission

(Invited Paper)

Sufian Hameed
University of Göttingen
shameed@cs.uni-goettingen.de

Xiaoming Fu
University of Göttingen
fu@cs.uni-goettingen.de

Pan Hui
Deutsche Telekom Labs
pan.hui@telekom.de

Nishanth Sastry
University of Cambridge
nishanth.sastry@cl.cam.ac.uk

Abstract—In this paper we introduce *LENS*, a novel spam protection system based on the *recipient's social network*, which allows correspondence within the social circle to directly pass to the mailbox and further mitigates spam beyond social circles. The key idea in *LENS* is to select legitimate and authentic users, called Gatekeepers (GKs), from outside the recipients social circle and within pre-defined social distances. Unless a GK vouches for the emails of potential senders from outside the social circle of a particular recipient, those e-mails are prevented from transmission. In this way *LENS* drastically reduces the consumption of Internet bandwidth by spam. Using extensive evaluations, we show that *LENS* provides each recipient reliable email delivery from a large fraction of the social network. We also evaluate the computational complexity of email processing with *LENS* deployed on two Mail Servers (MSs) and compared it with the most popular content-based filter i.e SpamAssassin. *LENS* proved to be fast in processing emails (around 2-3 orders of magnitude better than SpamAssassin) and scales efficiently with increasing community size and GKs.

I. INTRODUCTION

Spam emails are still an open problem largely outnumbering legitimate ones. In 2010, 89%[1] of the emails were spams (262 billion spam messages daily) and the projections show that spam will incur a cost of $338 billion[2] by 2013.

The common state-of-the-art strategy used today only filter spam from the user's inbox (i.e. recipient's edge), but the spam already travels the network, and provokes non-negligible cost to network operators in terms of bandwidth and infrastructure. On the other hand, content-based filtering [2] has turned spam problem into false +ve and -ve one.

There have been innumerable attempts to solve the problem of spam, including, recently, solutions that exploit trust embedded in social networks to create solutions without false positives [5], [9]. RE: [5] introduced the idea that recipients can trust senders in their immediate social neighbourhood, and gave a zero false-positive mechanism for vetting emails sent by their immediate social circle i.e. direct friends and friends of friends (FoF). However, email coming outside this circle still had to be tested by noisy and unreliable spam filters.

In this work, we aim to create a system that, like RE:, can be deployed individually by small groups of users, but

[1]http://royal.pingdom.com/2011/01/19/email-spam-statistics/
[2]http://www.redcondor.com/company/

allows for a reach greater than FoF. In order to accomplish this, we create a per-recipient ego-centric view of the entire social network of email users. Anyone who is a friend or FoF can email the recipient directly. To enable legitimate senders who are farther away, the recipient enlists a set of trusted users, called Gate Keepers (GKs) at various hop counts away from himself. Each GK is allowed to vouch for new senders in his immediate social circle by issuing them unforgeable vouchers. Unless a GK vouches for the emails of potential senders from outside the social circle of a particular recipient, those e-mails are prevented from transmission. In this way *LENS* drastically reduces the consumption of Internet bandwidth by spam to control messages only.

We further demonstrate the scalability of *LENS* in term of the number of email users using Facebook dataset. We show that with the help of hundreds of GKs, a recipient can be possibly reached by millions of users. The solution can be scalably extended to users with larger social distances by iterative GK selection. We also evaluate the computational complexity of email processing with *LENS* deployed on two MSs of different processing powers. We demonstrate that *LENS* is quite fast in processing emails and also compared its performance with the most popular content-based filter i.e SpamAssassin [2]. *LENS* also scales efficiently with increasing community size and GKs with computational overhead of couple of milliseconds.

The rest of the paper is organized as follows. §II describes the state of the art. §III describes the design of *LENS*. In §IV we discuss how *LENS* is realized and incorporated with existing email processing system. In §V we demonstrate scalability, effectiveness and complexity of *LENS* using online social network (OSN)/email datasets and system evaluations. Finally we conclude the paper in §VI.

II. RELATED WORK

Recently, several techniques [3]–[6], [8], [9] have been proposed using social networks and trust and reputation systems to combat spam. Boykin *et al.* create a social network of friends in the cyberspace based on the emails exchanged between them [3]. Using local clustering properties of social network they are able to classify 53% of all emails as spam or non-spam with 100% accuracy. However, the method is

limited to offline analysis, and the remaining 47% emails are left for other filtering techniques.

Ostra [9] utilizes trust relationship to thwart unwanted communication, where the number of a user's trust relationships is used to limit the amount of unwanted communications he can produce. *Ostra* relies on existing trust networks to connect senders and receivers via chains of pair-wise trust relationship and use a pair-wise, link-based credit scheme to impose a cost on originator of unwanted communication. However, its scalability of this system stays uncertain as it employs a per-link credit scheme.

Re: Reliable Email [5] exploits the use of white list of friends and automatic white list of FoF to increase the communication chance of only white list friends. By using this protocol, RE can accept almost 85% of received emails and prevent up to 88% false positive by the existing spam filters. However, emails other then friends and FoF still had to be filtered by noisy and unreliable spam filters. *LENS*, on the other hand, extends the reliable delivery of emails beyond FoF with the help of legitimate GKs.

SOAP [8] presents a social network based personalized spam filter that integrates social closeness, user (dis)interest and adaptive trust management into a Bayesian filter. However, several issues with SOAP, including the intrinsic cost of initialization and continuous adaptation of social closeness (between sender and recipient) and social interests (of an individual) in the Bayesian filter, limit its usage.

SocialFilter [10] proposes a collaborative spam mitigation system that uses social trust embedded in OSN to asses the trustworthiness of Spam reporter. The spammer reports from the SocialFilter nodes are stored at a centralized repository that computes the trust values of the reports and identifies spammers based on IP addresses. However, the SocialFilter's effectiveness is doubtful as spammers may use dynamic IPs.

In *Trust and Reputation Systems*, network users try to calculate the reliability and trustworthiness of other users based on their own experiences and that of others. Boykin *et al.* [3] proposed an automatic email ranking system based on trust and reputation algorithms. *MailRank* [4] is a spam detection system based on trust and reputation scheme to classify email addresses (apart from ranking emails as done in [3]) into spammer addresses and non-spammer addresses. It additionally determines the relative rank of an email address with respect to other email addresses. SNARE [6] infers the reputation of an email sender solely based on network-level features, without looking at the contents of a message. Using an automated reputation engine, SNARE classifies email senders as spammers or legitimate with about a 70% detection rate for less than a 0.3% false positive rate. However, lacking authentication and non-repudiation in standard trust and reputation solution make these solutions be subject to identity spoofing, false accusation and collusion attacks. Further, these solutions consume extra valuable resources of email servers on email reception and filtering. In contrast, *LENS* can reject unwanted email traffic during the SMTP time.

III. *LENS* ARCHITECTURE

LENS (cf. Fig. 1) comprises of four main components: 1) community formation, 2) trust management, 3) GK selection and 4) spam report handler. All the components of *LENS* run on a Mail Server (MS) along with the Mail Transfer Agent (MTA) and SMTP server. Abbreviations used in this section and elsewhere (to save space) are listed below for reference:

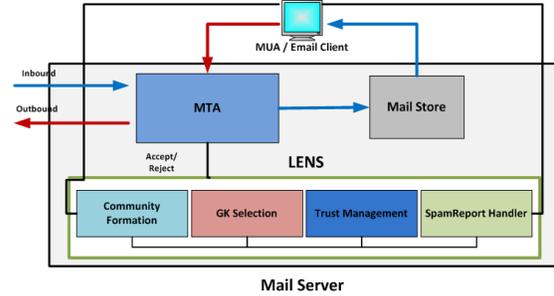| MS	Mail Server	GK	Gatekeeper
RN	Recipient node	BN	Boundary Node (FoF)
PK	Public Key (RSA)	SK	Secret Key (RSA)

Fig. 1. *LENS* Architecture

In *LENS*, the MS is responsible for executing the protocol on behalf of the email users. Each email user can explicitly control his community (friends and FoF) and can give feedback by reporting spam emails. All the remaining functionality of *LENS* is handled transparently by the MS.

All the *LENS* enabled MSs are assumed to be legitimate with a valid certificate issued from a *Trusted Authority*. These certificates are used during server authentication to prove that the MS is legitimate. All the authentication requests associated with invalid certificates are ignored. When a MS is certified (like any web-server) by a *trusted third party*, it shows that the MS belongs to a legitimate owner. The owner who controls the MS becomes visible, allowing legal actions to be taken against the owner or blacklist the MS.

Adding email users is strictly moderated in companies, private institutes and universities. Furthermore, all the major web-mail providers run bot detectors against non-human automatic account creation and impose an email sending limit (100 to 1000 recipients per day) [1]. However, illegitimate users can create accounts on web-mail providers like gmail, yahoo or hotmail. *LENS* addresses this problem by maintaining a trust rating for each user to ensure their legitimacy and differentiate between legitimate and illegitimate ones.

A. Community Formation

The formation of a social community is a simple two step process i.e. *addition of friends and FoF*. For FoF addition friend lists are not exchange among the friends. Instead a user can suggest two of his friends to add each other as FoF. By design, community formation is a selective process to preserve privacy.

B. Trust Management

This component is responsible for maintaining a system wide trust rating (TR) of the users and use them to determine the user type i.e. *legitimate, new and illegitimate*. Calculation of TR use mechanisms that are in principle similar to MailRank [4].

C. Spam Report Handler

This component handles spam reports and weights them according to the TR of the user to identify potential spammer.

D. GK Selection

GK serve as a means to vouch for legitimate users outside the community of the recipient for communication. To maintain a reliable trust structure, a GK is only authorized to vouch for the nodes in his own community. The GK selection consists of three stages as follows. *LENS* covers all the communication scenarios for legitimate emails (no legitimate email is stopped from transmission).
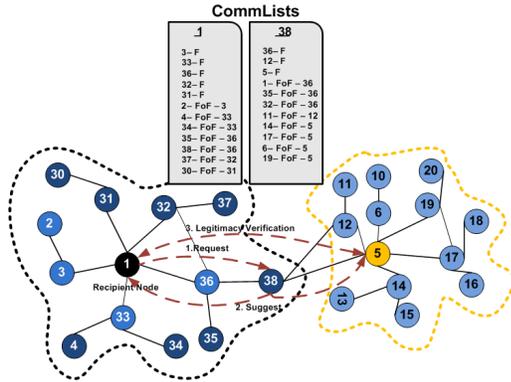

Fig. 2.   Community structure and GK coverage

1) Stage 1: GK selection in adjacent communities: The GK selection for a (RN) in adjacent communities consist of three step.

i) Request: RN's MS requests all the BNs i.e. FoF in the RN's community to send their suggestion for good GKs.

ii) Suggestion: The MS of the BN will suggest a locally optimal (highest degree) friend of the BN to the RN as a GK. The MS of the BN will also inform the BN's friends about the recipient. Exchange of friend list is not required for making suggestions, the MS can use the information in the BN's CommList [3]. For instance in Fig. 2, which depicts the selection of GK by the BN of the recipient, node "38" suggests "5" as the locally optimal GK to "1" instead of "12", since "5" is the friend with biggest community (and who is outside the community of "1"). The RN will choose the set of GKs that provide the best coverage. At the end of Stage 1, nodes within 5 hops of the RN can send emails to the RN.

---

[3]CommList is maintained for every user and it contains entries of community nodes, either as friend or FoF (see Fig. 2).

iii) Verification of Legitimacy: This is the last and most important step of GK selection process. This step ensures that the GK is legitimate. As a result of this step, a RSA based public key (PK) and secret key (SK) is generated for the GK. PK is shared with the RN and the SK is use to issue vouchers to entire community members of the GK. These members will use the issued vouchers if they need to communicate with the RN. All the users within a social radius (level or hops) of 5 would be able to send emails to the recipient with an assurance of being free from spam. Distant users having a social distance greater than 5 are covered in stage 2 of the GK selection process. PKList [4] and VoucherList [5] are maintained at each node to store the PKs and vouchers.
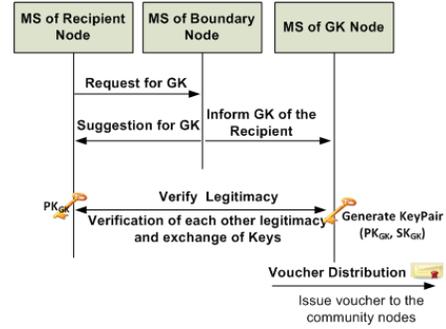

Fig. 3.   GK selection and voucher distribution

2) Stage 2: GK selection beyond adjacent communities: After completion of stage 1, RN's MS *sends a request to the selected GK's MSs* to help them look for GKs from their adjacent communities. As a result of this request, the GKs will use their BN to find new locally optimal GKs and *send their suggestions back to the RN*. Finally, *the RN's MS will verify legitimacy of the new set of GKs from social level 6* and extend reachability of the RN to level 8. The GK selection for higher levels must also consider the small world property of social networks [11], in order to *avoid random walks* on the social graph.

3) Stage 3: GK selection for new communication: If a user wants to send an email to a recipient (for the first time), who is not only outside its community but there is also no GK for the recipient within its community, *LENS* will perform the following two steps.

i) Announcement: announce the sender to the RN that wants to communicate.

ii) Verification of Legitimacy: start the legitimacy verification process to prove that the sender is not a spammer. As

---

[4]PKList is maintained for GKs selected by the recipients. Any single entry in PKList contains PK, GKID and RecipientID.

[5]Once the GK has the SK, a voucher $(((User_{ID})_{hash})_{Sign-SK})$ is issued to all the users of the GK's community. These vouchers are added to the VoucherList of the users, along with the recipient's and GK's IDs ($< RN_{ID1}, RN_{ID2}, ... >$, $GK_{ID}, Voucher[((User_{ID})_{hash})_{Sign-SK}]$),to use later for communicating with the recipient. A single voucher issued by the GK (to each of his community user) will work for all recipients.

a result of this process, the RN will add the sender as his GK.

This process is only performed once at the start of a new communication. After the sender is verified as a GK, not only the sender but his entire community can send email to the recipient.

4) Legitimacy verification in GK Selection: Legitimacy verification is a critical part of GK selection process. This protocol ensures that the recipient can trust the GK to be legitimate. With this protocol a RSA based PK and SK is generated for the GK. PK is shared with the recipient and the GK uses the SK to issue vouchers to his community members as a vouching mechanism to send emails to the recipient.

Legitimacy verification is two step process involving the verification of the GK and its MS. During the first step, the legitimacy of the GK's MS is verified against a valid certificate issued from a *Trusted Authority*. In the next step, a system wide trust rating *TR* of the GK from the trust manager is used. GKs with a TR above a certain threshold are considered to be legitimate.

In short, if a GK belong to a legitimate MS and has a good TR it is considered to be legitimate as well. Here we would like to emphasize that the main focus of *LENS* is spam protection, so only verifying the legitimacy of a user is enough to counter spam, rather than running costly protocols to authenticate the identity of each user.

IV. EMAIL PROCESSING WITH *LENS*

In this section we discuss the processing of emails with *LENS* and how spam can be prevented from transmission.

When a mail is received, it can fall into one of several categories.

1. Message within the community: When a message is sent to any recipient within the community, the recipient's MS will verify the sender against the recipient's Commlist and place the message in the mailbox.

2. Message outside the community with GK: If a message is sent to a recipient outside the sender's community, the sender's MS will bind a voucher, issued by a authorized GK along with the message. On reception, the MS will verify the voucher using the PK stored in PKList against the GKID and place the message in the recipient's mailbox.

3. New Message outside the community without GK: If a new message is intended for a recipient outside the sender's community and with no voucher issued by any GK. The sender's MS will hold the message and start a GK selection procedure (stage 3). On successful completion, the sender will be selected as a GK for the recipient. The sender's MS will now bind a voucher with the withheld message and send it out. When the message arrives at the recipient's MS, the MS will verify the voucher using the PK stored in PKList against the GKID and place the message in the recipient's mailbox.

A. Prevention of Spam Transmission

One of the main contributions of *LENS* is that it prevents the transmission of spam across the network. Let us consider that the sender's and receiver's MS have already established a TCP connection. Now, when the sender's MS sends the RCPT TO command (RCPT TO:<forward-path>[SP <rcpt-parameters>] <CRLF>), it also appends the voucher and issuing GK's ID (for e.g RCPT TO:<example@abc.com>Voucher=1f2a91aa236d0012 GK=gk@example.com) as additional rcpt-parameters, to communicate with the recipient, if the recipient is not in the sender's community.

According to the current draft standard of SMTP [7] using additional rcpt-parameters is optional and contemporary SMTP implementations MUST support it as basic extension mechanisms. The SMTP server not obliged to understand the additional rcpt-parameters simply ignores them. At the recipient's end, the MS verifies if the sender is a community member or has a valid voucher from a authorized GK. Failure of the verification results in the termination of the TCP connection by the recipient and the transmission of email (header and body) will not take place, thereby preventing the spam message from being transmitted.

B. Forgery of from Addresses

In *LENS* we have authentication at the MS level to verify the legitimacy of the MS. Further, addresses are not authenticated in SMTP. Therefore, it will be very easy for the spammer to launch a spam attack with forged ***from*** addresses as if they are from the recipient's community. In order to solve this problem *LENS* utilizes SPF [13] as standard sender authentication techniques to robustly verify that the *from* address in the received email is not forged. SPF filters the inbound email at MAIL FROM: command and is already being used effectively in the existing email system.

V. EVALUATIONS

This section presents performance evaluation of *LENS* using trace-driven simulations (§V-A)and our Linux implementation of *LENS* (§V-B).

To study the **scalability** of *LENS* (§V-A), we developed simulations based on Facebook dataset. We focus on the GK selection procedure at stage 1 because of the limitation of the dataset size. Although the dataset contains millions of users, the average path lengths are no more than 5 hops.

TABLE I
HIGH-LEVEL STATS OF FACEBOOK DATASET

Social network data set	Facebook
Number of Users	3,097,165
Number of Edges	23,667,394
Average Friends	15.28
Clustering Coefficient	0.175
Avg Path Length	5.13
Average Community Size	1,587.32

Table I presents the high-level statistics of Facebook dataset gathered and used in [12]. Our data sample of

Facebook consists of 3.1 million users with over 23 million edges and an average of 15.2 friends per user.
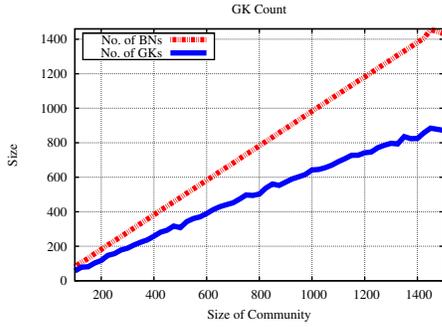
(a) No. of GKs
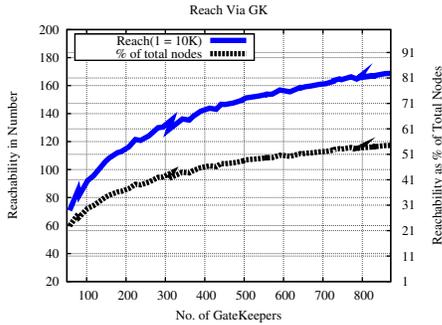
(b) Reachability via GKs

Fig. 4. Evaluation with Facebook Data

To evaluate the **computational complexity of email processing** with *LENS* (§V-B), we developed a *LENS* prototype implementation on Linux for mail processing during the SMTP transactions. We integrated this prototype with the mutt mail client, the Mail Avenger SMTP daemon (www.mailavenger.org), and the Postfix MTA (www.postfix.org). Without any modification to the SMTP implementation, *LENS* email filter (using CommList and vouchers issued by the GK) run as an independent daemon (like spamd for SpamAssassin: http://spamassassin.apache.org/) to monitor the SMTP transaction and based on the results take different actions. Community formation, GK selection and trust management run as independent components on the MS. The prototype uses SHA1 for hashing and RSA-based signature and verification.

A. Scalability Evaluation with Facebook Data

We randomly selected 4000 nodes from the Facebook dataset and tested them for GK selection in *LENS*. The nodes are selected randomly with the constraints in the community size ([100, 1500]) and the number of friends for any given node (>25). This is to mimic real social network size.

Number of GKs for receiving messages: Fig. 4(a) presents the number of GKs selected for a recipient to receive messages from outside its community. The number of required GKs is quite reasonable, ranging from 56 to 871 but mostly remaining less than half of the community size. The number of GKs show a nearly linear relationship with the number of boundary nodes. Increase in the number of boundary nodes results in a relative increase in the number of GKs but this is not always the case. It is also observed that a higher number of boundary nodes result in smaller number of GKs. The GK number is lower if the GK is selected from a region where the nodes have high clustering coefficient, which results in the suggestion of the same GK from a number of boundary nodes.

Reachability of recipient via GKs: Fig. 4(b) shows the number of users that can reach a particular recipient with the help of GKs. With a minimum number of GKs, the reachability of a recipient ranges from 700K to 1.7 million, or 22% to 55% of the total network, and remains over 40% most of the time.

The results suggest that *LENS* is scalable in terms of number of required GKs and reachability. With the help of only hundreds of GKs, a recipient can be reached by millions of users and the solution can be extended to the users with even further social distance by further GK selection. In contrast, RE: which handles up to FoF, will only be reachable reliably from his community (friends, FoF). An increase in the recipient's community size directly affects the reachability. Users having a larger community would benefit more from *LENS* than isolated and less socially connected users.

B. Computational Complexity of email processing with LENS

To analyze the computational complexity of email processing, we augment a standard mail processing system with *LENS* and measure its impact. For all the experiments, we used two MSs running Mail Avenger SMTP server on top of the Postfix MTA. The MSs are connected via a local area network. One of the MS is a intel atom 1.6 GHz with 1 GB ram and the other one is an intel core2duo 2.53 GHz with 4 GB ram. We measure the effect of community size, size of VoucherList and the number of GKs on the email processing using *LENS*. Note that in our experiments we only measured computational overhead (in terms of time) on the SMTP transaction (i.e. RCPT TO) monitored by *LENS*.

1) Effect of CommList at the recipient's MS: On receiving the RCPT TO command, the recipient's MS performs a check to see if the sender is in the recipient's community, and take actions (accept/reject/pass to other filters)according to the policy. In Figure 5(a), we present the results to show the overhead of community size on the email processing with *LENS*. The community size varies from 0 to 10K. The computation overhead on both the MSs is only in few milliseconds (ms). With the community size of 10K the overhead at one MS (intel atom 1.6 GHz) is 30 ms and around 9 ms on the other (intel core2duo 2.53 GHz).
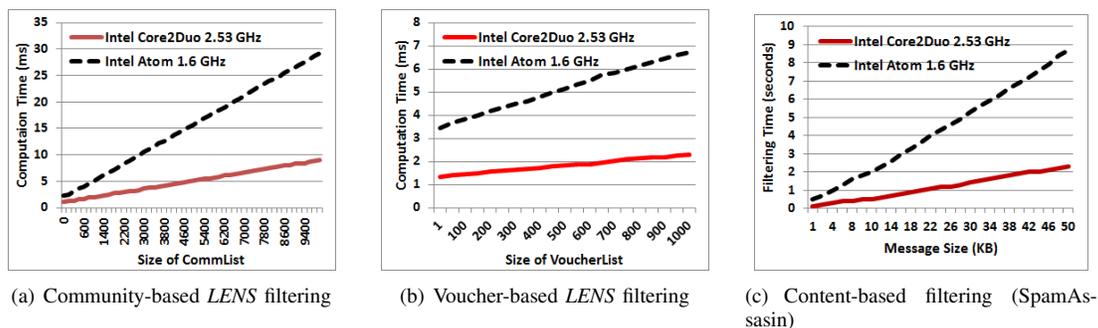
(a) Community-based *LENS* filtering (b) Voucher-based *LENS* filtering (c) Content-based filtering (SpamAssasin)

Fig. 5. Performance of email filteration (scales of a, b and c are not same)

2) Complexity of signature verification and effect of PKList at the recipient's MS: The cost of checking signature has been has been so far considered substantial. We would like to do some measurements to ensure that the proposed scheme, with its voucher verification does not cost more computation than processing spam. In order to study the computational cost of voucher generation/verification, we perform OpenSSL (www.openssl.org) speed measurement run on the servers. It takes approximately 0.13 ms to sign and 0.07 ms to verify a message using 1024 bits RSA and SHA-1.

Voucher verification, when integrated in *LENS* for email processing, depends on the size of PKList as well. Note that the size of the PKList of a recipient is the same as the number of its GKs. Figure 5(b) shows the overhead of PKList size on the voucher verification with *LENS*. The average GK count in Enron is 31, and in the worse case 235. Based on that we varied the PKList size between 1 to 1000 for the experiments. The computational overhead on both the MSs is only few milliseconds (ms). With the PKList size of 1000 the overhead at one MS (intel atom 1.6 GHz) is 6.7 ms and around 2.3 ms on the other (intel core2duo 2.53 GHz).

3) Comparison with content-based filtering: SpamAssassin, one of the most widely used content-based filter, uses a variety of mechanisms including header and text analysis, Bayesian filtering, DNS blocklists, and collaborative filtering databases to filter spam before it reaches the mailbox. Figure 5(c) shows the computational time taken by SpamAssassin to filter messages of different sizes. We varied the message size from 1 KB to 50 KB and analyzed the performance of SpamAssassin. SpamAssassin takes 0.1 sec on intel core2duo 2.53 GHz machine and 0.5 sec on intel atom 1.6 GHz for a message size of 1 KB. SpamAssassin is linear to the size of the message and the filtering time increases with an increase in the message size. SpamAssassin takes 2.3 sec on intel core2duo 2.53 GHz machine and 8.7 sec on intel atom 1.6 GHz for a message size of 50 KB. On the other hand, *LENS* is totally independent to the message size.

Based on the results presented in this section, we conclude that *LENS* is computationally efficient. *LENS* is fast in processing emails and it is 2-3 orders of magnitude faster than SpamAssassin. It also scales efficiently with increasing size of the lists (CommList, VoucherList and PKList), and even on a low end processing machines the overheads are in couple of milliseconds.

VI. CONCLUSION

In this paper we introduce *LENS*, a novel spam protection system based on the *social networking paradigm*, which mitigates spam beyond recipient's social circles with the help of trusted users, called GKs. *LENS* covers all the communication scenarios for legitimate emails (no legitimate email is stopped from transmission).

Our initial evaluation results, based on Facebook traces, demonstrate that reliable email delivery from millions of potential users is possible using GKs in the order of hundreds. Initial evaluations of the system prototype reveals that *LENS* remains lightweight and performs significantly better than SpamAssassin [2]. More experiments are being performed to evaluate system perform under various scenarios for further potential improvements.

REFERENCES

[1] Email address limit in webmail by providers. http://www.emailaddressmanager.com/tips/email-address-limit.html.
[2] Spamassassin. http://spamassassin.apache.org/.
[3] P. O. Boykin and V. Roychowdhury. Personal email networks: An effective anti-spam tool. *IEEE COMPUTER*, 2004.
[4] Paul Alexandru Chirita, Jörg Diederich, and Wolfgang Nejdl. Mailrank: using ranking for spam detection. In *Proc. of CIKM*.
[5] S. Garriss, M. Kaminsky, M. J. Freedman, B. Karp, D. Mazi'eres, and H. Yu. Re: Reliable email. In *Proc. of NSDI*, 2006.
[6] Shuang Hao, Nadeem Ahmed Syed, Nick Feamster, Er G. Gray, and Sven Krasser. Detecting spammers with snare: Spatio-temporal network-level automatic reputation engine. In *USENIX Security*, 2009.
[7] J. Klensin. Simple mail transfer protocol, j. klensin. The Internet Society, RFC 5321, October 2008.
[8] Z. Li and H. Shen. Soap: A social network aided personalized and effective spam filter to clean your e-mail box. In *Proc. of IEEE INFOCOM*, 2011.
[9] A. Mislove, A. Post, P. Druschel, and KP Gummadi. Ostra: Leveraging trust to thwart unwanted communication. In *Proc. of NSDI*, 2008.
[10] M. Sirivianos, K. Kim, and X. Yang. Introducing social trust to collaborative spam mitigation. In *Proc. of IEEE INFOCOM*, 2011.
[11] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 1969.
[12] Christo Wilson, Bryce Boe, Alessandra Sala, Krishna P.N. Puttaswamy, and Ben Y. Zhao. User interactions in social networks and their implications. In *Proc. of EuroSys*, 2009.
[13] M. W. Wong. Sender authentication: What to do. http://spf.pobox.com/whitepaper.pdf, July 2005.