Tracking Email Reputation for Authenticated Sender Identities

Gautam Singaraju, Jeff Moss, and Brent ByungHoon Kang
College of Computing and Informatics, University of North Carolina at Charlotte

Abstract

With the amount of unsolicited emails on the rise, domain authentication schemes have been widely deployed to identify senders. Establishing a sender's identity does not guarantee its adherence to best practices. To maintain a history of sender activity, in our prior work, we had proposed RepuScore: a collaborative sender reputation framework and demonstrated its effectiveness using simulated logs. In this paper, we share our initial experience in deploying RepuScore along with its SpamAssassin plug-in that we recently developed. From the deployed RepuScore, we learned that a variation in the received email volume disguises the real reputation of the sender. To solve this problem, we propose Volume-Enhanced RepuScore that, in addition to spam rate, considers the received email volume. Based on the deployment since 10/9/2007 at two organizations, we show that: a) RepuScore data can be used to identify authenticated senders and classify their emails. Using the computed reputations for 23 days, RepuScore classified emails from about 42% of the authenticated sender identities that correspond to about 72% of the authenticated email volume; b) sender identities with low reputations have a shorter lifetime compared to ones with high reputations.

1. Introduction

Unsolicited email, popularly referred to as spam, has grown to epidemic proportions. Spam presently contributes to about 90% of all email on the Internet [18] and estimates place the financial losses due to Phishing around $2.8 billion a year [16]. Sender Identity [20] is one of the proposed techniques to verify senders before accepting their emails. Sender Identity has rapidly come to the forefront with advent of DKIM [1, 20], DomainKeys [29], SPF [28] and SenderID [17]. A recent study shows that about 35% of all email is authenticated [14].

Unfortunately, a sender's identity alone does not necessarily guarantee their adherence to best email practices. For example, from our experiments (described in Section 4), we noticed that about 89% of the authenticated sender identities were spammers. As sender identity takes center-stage, these observations motivate maintenance of a long history of sender activity to classify their emails.

Our previous work, RepuScore [24] is a collaborative reputation framework where the receivers report their reputation-view about sender to a central authority that computes a global reputation for each of the sender identities[1].

The RepuScore framework was designed to place the onus on the senders to control the amount of unsolicited emails they transmit. By using the globally-computed quantitative scores, receivers can select a minimum reputation to accept emails from the authenticated sender identities. Such a mechanism makes it costly for spam propagators to reach inboxes of actual users.

Our previous work demonstrated a proof-of-concept design and evaluated the algorithm using simulated logs. The RepuScore algorithm was demonstrated to be secure against Sybil attacks [9, 26, 32], where malicious senders control multiple identities (Sybils), each of which is used to increase reputation of the attacker's own identities.

Since our previous work, RepuScore has been deployed at two receiver organizations[2] (from 10/9/2007 – to present). During this time, reputations for over 16,500 sender identities have been computed.

This paper offers discussions on challenges we faced and the results of our deployment:

a) In the original algorithm, while using the spam-rate from each sender identity, variations in the received email volume significantly impacted the reputation. We propose *Volume-Enhanced RepuScore,* an extension to the algorithm that now uses both spam-rate and email volume for computing reputation.

b) We discuss the design of a RepuScore plug-in for SpamAssassin. The plug-in uses existing SpamAssassin plug-ins to verify the sender identity and transmits the information to a local *RepuServer*.

c) We share our observations and statistics about the RepuScore:
   i. With knowledge of only 42% of the sender identities, RepuScore classified 72% of the received emails. About 11% of the sender identities were good, while 32% were spammers.

[1] Henceforth, sender identity will be used to denote **authenticated sender** including sub-domains.

[2] Henceforth, receiver organization will be used to denote organizations that share reputation information.

ii. 97.8% of the sender identities had reputation either near 0 or near 1.

iii. Sender identities with low reputation have a shorter lifetime as compared to ones with high reputation. Average lifetime of good and bad sender identity was 61.9 and 17.47 days respectively.

iv. A large number of sender identities are created constantly that sent emails only in a single interval.

v. Reputation can be accurate in determining if a sender identity is a spammer.

This paper has been organized as follows: Section 2 discusses the related work, and Section 3 describes our deployment at a single receiver organization. Section 4 presents our results and we finally conclude in Section 5.

2. Related Work and Background

In this section, we discuss sender identity frameworks, followed by existing reputation management frameworks for email and peer-to-peer networks.

Sender Identification techniques

To identify a valid sender, sender identification techniques such as SenderID [17], Sender Policy Framework (SPF) [28], Domain Key Identified Mail (DKIM) [1, 20] and DomainKeys [29] have been developed. SPF verifies if the sender's email server is authorized to send email for the sender's domain. SenderID differs from SPF in its ability to authenticate either the "envelope from" header or the "purported responsible address". Using DKIM and DomainKeys, senders publish a set of public keys as a part of DNS; emails are then signed by their mail server. Receivers verify the signature, and hence, the sender. Accredited DomainKeys [10] suggests the use of a central server that monitors the senders' activity to evaluate them. Reputation can be one way to monitor the sender activity.

A recent study reports that 35% of all email is authenticated using one of the sender identity techniques [14]. Reputation frameworks based on a sender identity technique can reliably classify senders to maintain a reliable history of conformance to a common guideline.

Reputation Management for Email Infrastructure

Email reputation can use either the senders' IP addresses or their domain name as a basis to assign reputation.

SenderPath's Sender Score [22] and Habeas' SenderIndex [11] provide reputation for a sender's IP address. SecureComputing's TrustedSource [7] provides a global reputation system with the help of their deployed mail servers across multiple organizations. These organizations focus on creating a set of bad sender IP addresses (*not domain names*) to reject emails from them.

To create a group of senders whose prolonged history vouches for its email best practices, a reputation management system should use a domain name rather than the sender domain's IP addresses. Basing reputation on the domain name strongly ties an organization with its past email activity because (i) an IP address does not intuitively translate to a domain name [8]; (ii) multiple organizations can share an IP address; (iii) credible organizations in general would maintain their domain name for a longer period of time than their IP addresses.

Project Lumos [13] was proposed as an effort to provide reputation among collaborating ISPs. However, the project does not seem to be actively deployed yet. Project Lumos was intended to provide a receiver feedback to determine if the sender is a spammer or otherwise. It suggested reputation based on the weighted average of the past and present reputation views. Project Lumos considered all reputation reporters to be genuine, and therefore, did not consider attacks, such as Sybil Attacks, from reputation submitters.

Cloudmark's Network Classifier [21] is a community-based filter-system where multiple agents submit feedback about emails to nomination servers which require multiple users to confirm the claim that an email is spam. This information is submitted to a central server known as the Trust Evaluation System which computes a global view for an email's fingerprint. The Cloudmark paper advises not to use authenticated domain name as a fingerprint, as this would lead to a high multiplicity and cross-collision rate. In RepuScore, instead of using a fingerprint, we use the authenticated domain name to maintain reputation for each sender.

Google's reputation system [4] identifies senders using best-guess SPF or DKIM and computes the sender's reputation based on user inputs. Google's reputation system demonstrates high accuracy in classifying their emails. The author points out the need for a third party cross-domain collaborative reputation framework.

SenderPath's SenderScore Certified [23], Habeas' Safelist [12] and Goodmail's Certified Email [5] are certification and accreditation services that allow bulk senders to obtain third party certification. These systems do not qualify as reputation systems, as the senders control their own reputation rather than the receivers assigning reputations for them.

Reputation Management in Peer-to-Peer systems

Reputation management techniques have been used in agent based systems [25, 27] as a mechanism to evaluate trust. In multi-agent systems, peers use reputation to evaluate other agents to be able to select the best course of action to maximize their outcome [19]. Reputation systems have been prone to Sybil attacks [9] where a single attacker uses multiple identities to submit multiple reputation votes about its peers. Such attacks are detrimental to honest users and amicable to the attacker.

For all Sender Identities:

PresentRep = (Number of Good Emails)/ (Number of Emails)

If (ReportedRep (at Interval m-1) >= PresentRep)

ReportedRep (*at Interval m*) =

$\alpha \times ReportedRep\ (at\ Interval\ m\text{-}1) + (1\text{-}\alpha) \times PresentRep$

Else

ReportedRepu (*at Interval m*) =

$(1–\alpha) \times ReportedRep\ (at\ Interval\ m\text{-}1) + \alpha \times PresentRep$

α varies between (0, 1); α is the correlation factor that determines the importance placed in the past or present.

- α is closer to 0, the present interval is emphasized.
- α is closer to 1, the past interval is emphasized.

Equation 1: RepuServer reputation maintains history of spam rate as to maintain reputation.

Global Reputation (at Interval m) =

$$\frac{\text{Sum of \{RepuCollector's Repu (at interval m-1)} \times \text{reported Reputation by RepuCollector (at Interval m)\}}}{\text{Sum of All \{RepuCollector Repu (at Interval m-1)\}}}$$

Equation 2: Central Authority computes global reputation. To thwart Sybil Attacks, different weight is applied based on the RepuCollector's reputation.

To protect against deception and attacks in cooperative reputation systems, inputs from honest users are considered more valuable. The effectiveness of reputation protocols can be measured by their success in thwarting Sybil attacks. Using user personalized reputations in addition to the global reputations of the senders is one mechanism suggested to harden the reputation frameworks [6].

Using eigenvectors, EigenTrust [15] uses global reputation to identify malicious peers. Future reputation is calculated based on the present normalized trust reputation of all the peers. Due to the feedback mechanism, EigenTrust is a self-policing system that regards a trusted peer more than that of a peer of low-repute. Such a mechanism is helpful in guarding against Sybil attacks.

Another reputation framework has been developed as an application-independent system [30]. This system considers the multi-player prisoners dilemma, where every agent tries to maximize its own profits while maintaining the trust of other nodes. The system also incorporates [31] a mechanism to detect deceptions and reduce the effect of malicious votes from such peers.

RepuScore

Our previous work, RepuScore [24] is a reputation framework developed to incorporate inputs from both local users and peer receiver organizations to calculate global reputation for sender identities. Using such a collaborative scheme, organizations with relatively few users can classify emails from authenticated sender identities.

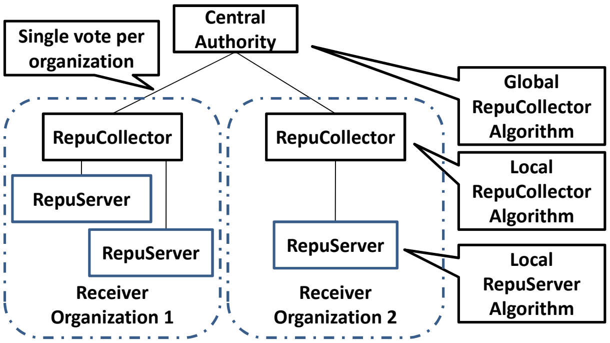


Figure 1: RepuScore Framework across receiver organizations. Each organization is allowed a single vote. Reputations is computed every Reputation Interval.

Since spammers frequently take new identities, a set of high-spam propagating sender identities cannot exist. In comparison, the group of non-spam propagating sender identities does not change frequently. Sender identities about which RepuScore does not have any information can be classified using other email classification techniques.

RepuScore introduces a central authority that collects reputation from multiple receiver organizations. RepuScore's centralized design enables receiver organizations to enforce reputations and remove malicious senders from a trusted group.

Figure 1 demonstrates the RepuScore architecture with the different entities, namely the RepuServer, the RepuCollector and the Central Authority, where:

a) RepuServers periodically compute reputations for sender identities as seen at the mail server;

b) RepuCollectors compute reputations for sender identities as seen from all mail servers at the organization;

c) Central Authority computes global reputation by combining scores submitted by all participating receiver organizations.

Equation 1 shows the algorithm used at a RepuServer. Using the Time Sliding Window Exponentially Weighted Moving Average (TSW-EWMA) algorithm [3], RepuScore maintains the history using spam-rate. Email reputation frameworks should include a feedback mechanism to compute reputation for entities [2]. The equation allows either a fast or a slow change (both increase and decrease) in the reputation. To provide an optimum behavior: slow increase but fast decrease, we interchange the weights α and 1 – α if the present reputation is greater than the one in the past.

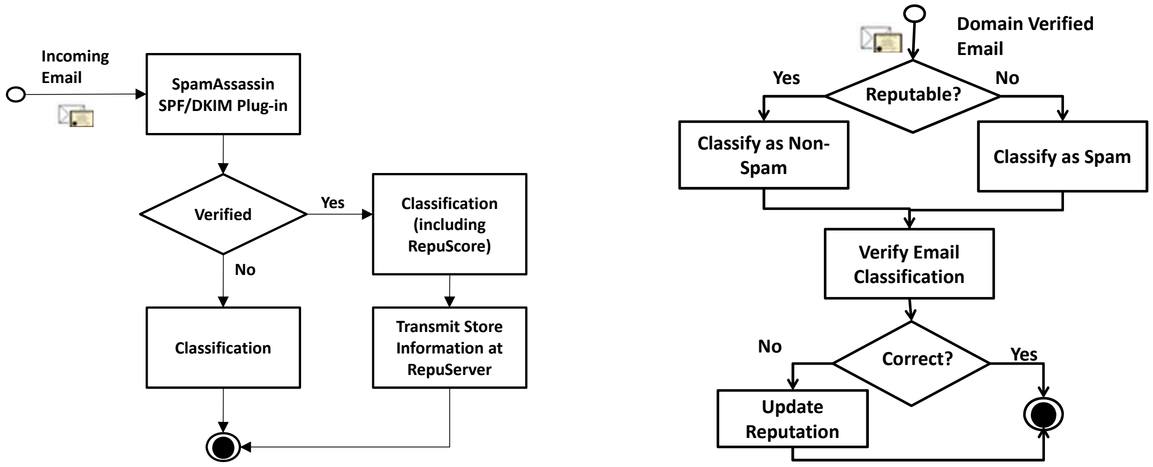


Figure 2 (A): SpamAssassin Plug-in collects statistics from mail servers and transmits it to a RepuServer. The plug-in uses other SpamAssassin plug-ins to identify sender identities. (B): Classification of email using the Plug-in. Email verification can use other spam classification techniques to correct the reputation-based classification.

Without the interchange of α and 1- α, for a large α, the reputation would increase and decrease quickly, whereas for smaller α, the reputation increases or decreases slowly. By interchanging the value of α and 1- α, for a large α, the reputation increases slowly and decreases quickly whereas for a small α, the reputation increases quickly and decreases slowly. The ideal behavior is to increase slowly and decrease quickly.

The local RepuServer transmits data to the RepuCollector to compute a local reputation. The RepuCollector averages all the RepuServer reputations and transmits it to a central authority. The Central Authority computes a global reputation based on the all the reported reputations from participating RepuCollectors. RepuScore handles Sybil attacks by valuing a reputable participant's rating more than that of a less reputable participant. RepuScore employed the Weighted Moving Algorithm Continuous (WMC) [30] to thwart Sybil attacks. Equation 2 demonstrates the reputation computation by the Central Authority received from different RepuCollectors.

3. RepuScore Deployment

In this section, we describe the deployment model. The gathered votes, from receiver organizations, are collected based on user inputs or email classification programs such as SpamAssassin.

RepuScore has been deployed at two receiver organizations since 10/9/2007, computing reputations for about 16,500+ sender identities.

3.1. SpamAssassin Plug-in

We developed a SpamAssassin plug-in that collects information about each authenticated email; i.e., whether or not an email is spam, and computes reputation for the sender identity. The RepuScore plug-in uses the available standard SpamAssassin plug-ins for SPF and DKIM to identify the senders.

Figure 2A demonstrates the design of the SpamAssassin plug-in that collects information for each email from an organization's mail server. After verification of the sender identity, the RepuScore plug-in classifies the sender. This process is further explained in Figure 2B. After sender verification, sender's reputation is used to classify the email. A reputable sender's email is classified as non-spam and vice versa. Reputation-based email classification requires a feedback mechanism for checking the accuracy of classification with the help of low-process intensive mail filters. As the RepuScore plug-in already has performed the sender identity checks, content-based filters can be utilized. The information is then transmitted as a UDP packet and stored at a local RepuServer.

System administrators can select any low-process intensive email classification technique to correct the information. Such a mechanism allows high-process intensive mechanism to be used to classify emails without an associated sender identity. This allows a faster email classification when a huge volume of email is received.

The RepuServer's server module (a Perl module) maintains multiple forked instances to keep a few "hot" instances in memory to handle the normal load, while having the ability to fork a few additional instances based on the need. These processes capture the packets transmitted to them by the RepuServer client module and write the incoming data into a MySQL database. A cronjob initializes a script that computes the reputation at every reputation interval by invoking SQL statements.

3.2. Volume-Enhanced RepuScore Algorithm

An interesting experience from our deployment was that the reputation of certain sender identities did not reflect the change in the email volume received from them. A

Given:

PastVol: Past Volume, PresVol: Present Volume, α’: a default value for α, GRpast: non-spam rate in the past interval, and, GRpres: non-spam rate in the present interval, Vol-Enh GR: Volume-Enhanced Good Rate:

```
If (PresVol > PastVol)
    Vol-Enh GR= (PastVol/PresVol) × GRpast + 1 × GRpres
Else
    Vol-Enh GR= 1 × GRpast + (PresVol/PastVol) × GRpres
End if
If (PresVol == PastVol)
        Instantaneous value of α = α’
Else
        Instantaneous value of α =
            $e^{(- \text{multiplicative factor} \times \text{Vol-Enh GR})}$
End if
```

where multiplicative factor is a constant used to decrease the large values of Vol-Enh GR.

Equation 3: **Instantaneous value of α based on the volume of email received at a RepuServer.**

constant spam rate does not imply that the volume of email is constant. For example, consider a spammer who propagates 1 spam email out of 10 emails in the first interval (spam rate = 0.1, reputation of 0.9) followed by 900 spam messages out of 1000 emails (spam rate = 0.9; reputation = 0.1) in the second interval. In this case, with a value α as 0.5, the reputation would be 0.5 (an average of 0.1 and 0.9). However, such a sender should be penalized more.

To track sender’s reputation more closely, more emphasis should be placed on the interval in which the email volume was higher. For example, if the email volume in the past interval was higher than the email volume in the present, more emphasis should be placed in the past. Likewise, when the email volume in the present is higher, the present reputation should be considered more than the past reputation.

Incorporating the change in the email volume on a global scale requires all the RepuCollectors to share both peer-reputations and the email volume. Sharing of the email volume invokes further attacks on the reputation framework; for instance, some receiver organizations could provide incorrect volume information about sender identities to increase/decrease their reputations. Our initial deployment showed that the majority of sender identities were spammers. As incorporating email volume at a global level, participating receiver organizations could lie about the volume sent by a sender. Because of this reason, email volume should be incorporated at RepuServers but not at the RepuCollectors.

Any incorrect volume embedded into reputations at RepuServer would only be constrained to the organization. Such incorrect reputation-view from a receiver organization will not significantly affect the global reputation since such data will be negated by other honest receiver organizations.

To incorporate email volume as a basis for the computation, we select exponentiation due to its monotonic property[3]. Due to this property, the e^{-x} always lies in the interval (0, 1) and is a monotonically decreasing function. A monotonically decreasing function is required as the value of α should decrease as the volume rate increases.

Equation 3 demonstrates our mechanism to compute the instantaneous correlation factor α based on the email volume. The Volume-Enhanced Good Rate (Vol-Enh GR) is the sum of the good rate in the interval that had larger volume and a fraction of the good rate in the other. This implies that having the Good Rate (GR) constant, if the volume in present is large, the Vol-Enh GR is the sum of good rate in the present and a fraction of the good rate in the past. If the volume in the past is small, the good rate in the past is small multiplied by the factor past volume divided by present volume. This leads to a lower value of Vol-Enh GR relative to the good rate in the present interval, leading to a higher instantaneous α. High α implies a more importance is placed on the past interval as compared to the present interval. Likewise, high Vol-Enh GR leads to a lower value for instantaneous α. The multiplicative factor is used to decrease large values of Vol-Enh GR.

In the same example discussed in the first paragraph, the Vol-Enh GR = 0.91. Using the multiplicative factor of 1, the Volume-Enhanced reputation will be 0.42 instead of 0.5.

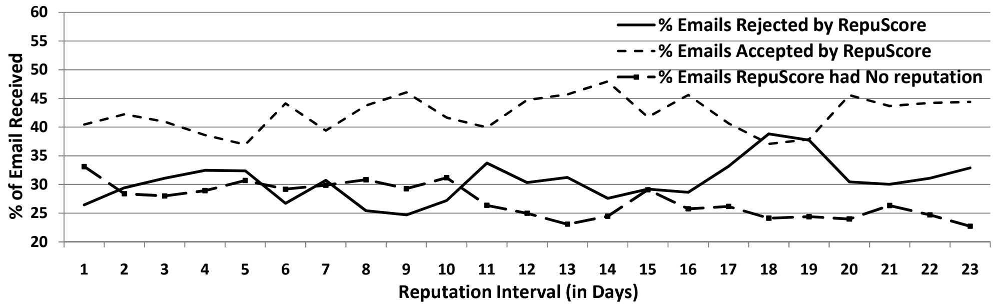
4. Results from our Deployment

In this section, we discuss the results of the deployment at two receiver organizations. We show the RepuScore statistics, effectiveness of RepuScore and the results of Volume-Enhanced RepuScore.

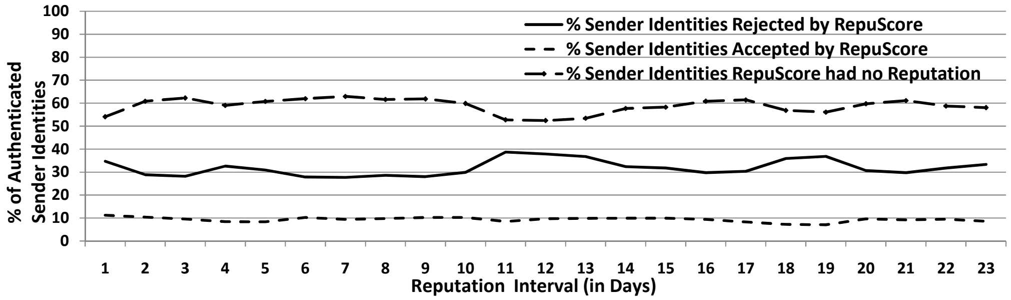
4.1. RepuScore Statistics

In our deployment for 174+ days, we computed reputations for 16,509 sender identities authenticated using SPF and DKIM. We define *Minimum Good Reputation* as the minimum reputation to be considered a credible sender. We select a value of 0.5 to classify the emails and discuss the reasons for selecting the same. We define *Lifetime* of a sender identity as the number of reputation intervals between the first and the last occasion including the first occasion the sender identity sent an email. For example, if the sender appears just on one day, the Lifetime is considered 1. We selected the value of α as 0.8 for original RepuScore for all comparisons. We select 0.8 to place more importance in the past than the present. The value of α’ was 0.8 to compare Volume-Enhanced RepuScore with RepuScore.

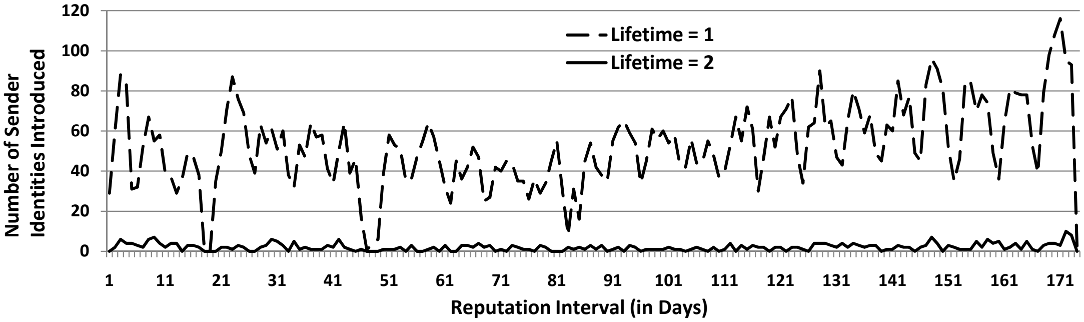
[3] A function, *f*, is called monotonic when given two non-distinct values a, b such that a > b, then f(a) > f(b).



Graph 1: Percentage of Authenticated Emails classified using RepuScore by evaluating mail logs of receiver organization 2. Reputation was computed from the receiver organization 1. On the average, RepuScore classified about 72% of received emails and accepted 40% of the emails. 32% of the emails were rejected.



Graph 2: Around 10% of the authenticated sender identities were credible senders; while about 32% were known spammers. RepuScore had no reputation information for 58% of the senders.



Graph 3: Number of sender identities with lifetime of 1 day (sent emails only on 1 day in 174 days) and 2 days (sent emails on 2 consecutive days in 174 days) plotted against their first appearance. We note that about 8000 new sender identities sent email only 1 time in 174 days.

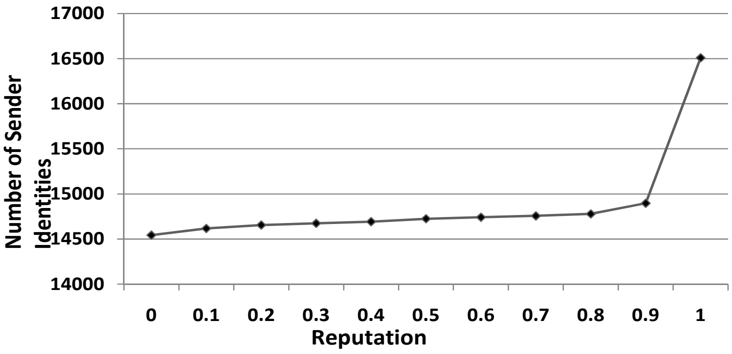
Our deployment experiments show information from two organizations. The first receiver organization is a small business organization with a user base of 50 that uses SpamAssassin to classify the emails. This organization has deployed RepuScore since 10/9/2007.

The second receiver organization is an ESP that has deployed RepuScore since 2/7/2008. The organization has 78,000 users of which about 10,000 paying customers have SpamAssassin plug-in to identify senders. About 17,000+ verified authenticated emails are received by the organization in a single day.

4.2. Effectiveness of RepuScore

To show the effectiveness of RepuScore, we use the reputation computed from the first receiver organization from day 107 to the mail logs from the second organization. In these graphs, the second organization uses SpamAssassin and not RepuScore to classify emails.

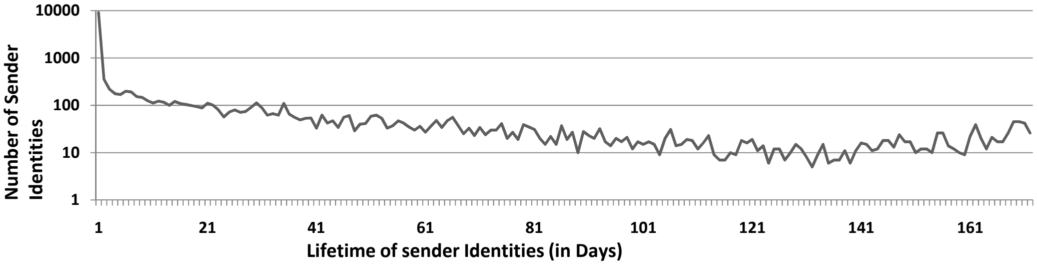
Graph 1 and Graph 2 shows the effectiveness of RepuScore in classifying authenticated emails. Our results show that using RepuScore, while only 10% of the sender identities were good over 23 days they transmitted about 40% of the authenticated emails. This 40% of emails were accepted by RepuScore. About 32% of the sender identities were spammers who sent about 32% of the authenticated emails. This 32% emails were rejected by RepuScore. Based on the information from Graph 1 and Graph 2, we infer that with the knowledge of about 42% (10 + 32) of the sender identities, RepuScore classified about 72% (40 + 32) of the authenticated emails.



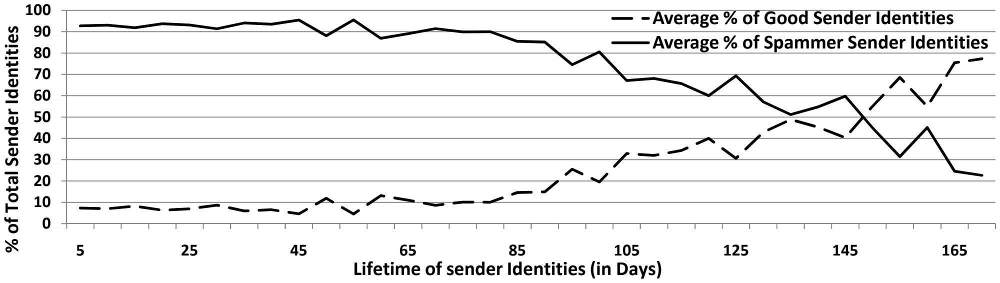
Graph 4: The cumulative distribution of sender identities as a function of reputation. 97.8% of the identities had a reputation of near 0 or near 1.

Minimum Good Reputation	Number of Good Domains
0 (From 0 to 1)	16,509 (100%)
0.1 (From 0.1 to 1)	1,925 (11.66%)
0.2 (From 0.2 to 1)	1,858 (11.25%)
0.3 (From 0.3 to 1)	1,834 (11.11%)
0.4 (From 0.4 to 1)	1,817 (11.01%)
0.5 (From 0.5 to 1)	1,803 (10.92%)
0.6 (From 0.6 to 1)	1,767 (10.70%)
0.7 (From 0.7 to 1)	1,752 (10.61%)
0.8 (From 0.8 to 1)	1,730 (10.48%)
0.9 (From 0.9 to 1)	1,681 (10.18%)
1 (Reputation of 1)	1,541 (9.33%)

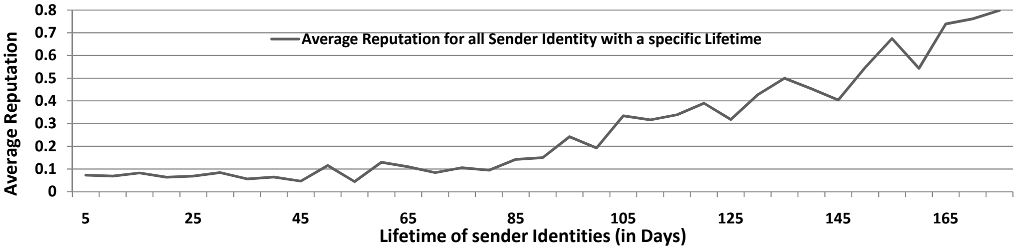
Table 1 shows the values by distribution against the minimum good reputation.



Graph 5: The distribution of the number of identities vs. their lifetime. The distribution of sender identities decreases as the lifetime increases.



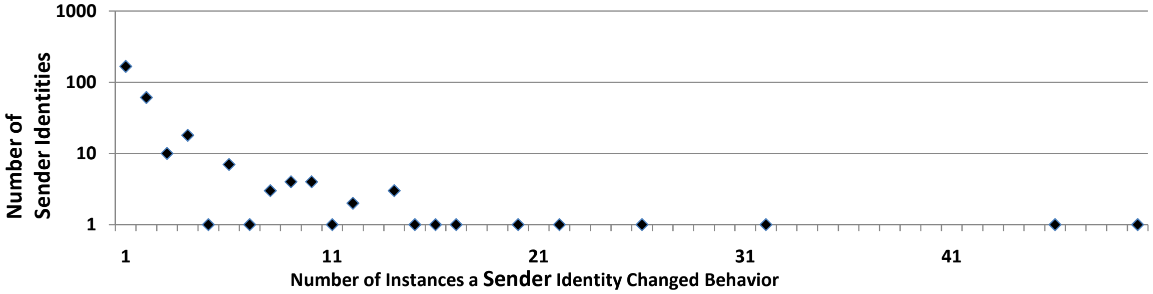
Graph 6: Percentage of good (or bad) sender identities to total number of sender identities as plotted against lifetime. The probability that a sender identity being credible increases with long lifetime.



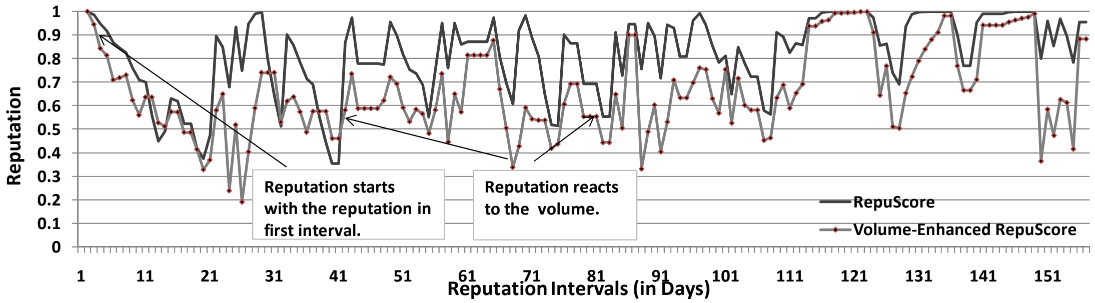
Graph 7: Average reputation of all sender identities with the same lifetime. As the lifetime increases, sender identity with longer lifetime has a higher reputation.

The results show that reputation gathered from a small set of users can be effective to classify emails for a large number of users. We noticed that the number of identities RepuScore had no knowledge about was always constant indicating that a lot of new one-time sender identities were being introduced.

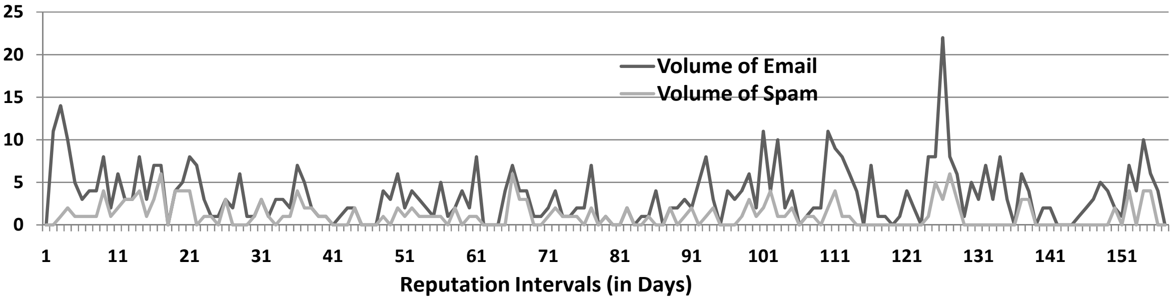
Graph 3 proves the hypothesis about a huge number of sender identities being created to spam and are taken down soon. We notice that sender identities with a lifetime of 1 day are distributed over the time of the deployment. The total number of identities that sent emails only in 1 interval was about 8000. The rate at



Graph 8: Number of times a sender identity changed from good to bad or vice-versa. Only 290 sender identities (about 1.75%) changed its behavior.



Graph 9: Volume-Enhanced RepuScore reacts based on the email volume for a popular free email provider. After volume enhancement, the reputation between the intervals 1-11, drops radically. Reputation increases quickly between the intervals 11-15. The slope is indicative of email volume in volume enhanced reputation.



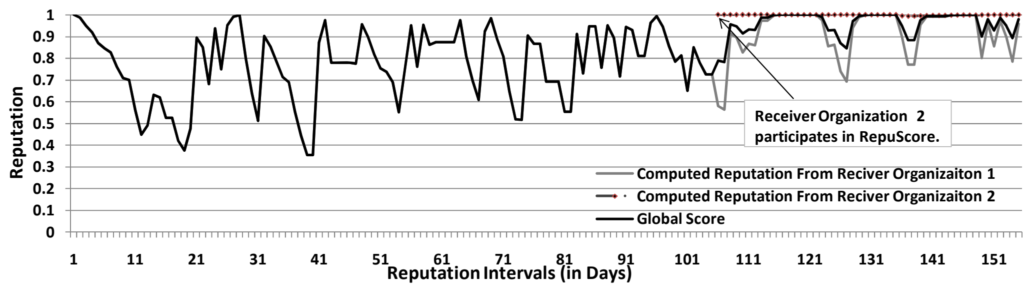
Graph 10: Volume of the Spam and Email noticed at receiver organization 1. The reputation of the sender identity changes based on the volume of reputation.

which identities sent email on two consecutive days was much lesser.

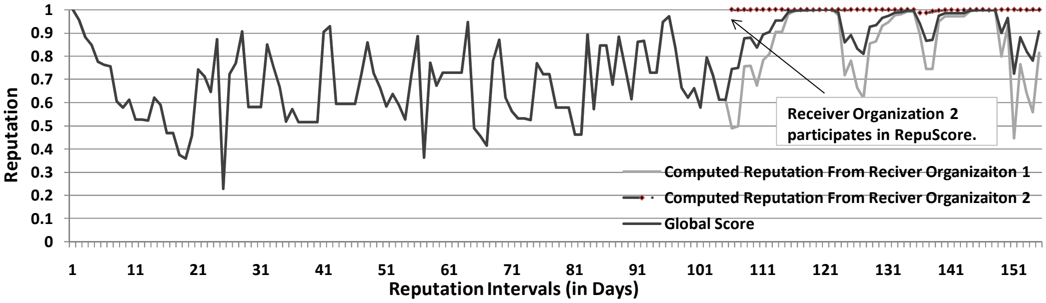
The cumulative distribution of sender identities as a function of reputation is demonstrated in Graph 4. Out of the 16,500+ identities, about 14,000 had a reputation of 0. The graph shows that about 97.8% of the senders have a reputation either around "0" or "1". With the help of Table 1, we select a minimum good reputation to be 0.5, the median value. With minimum good reputation as 0.5, only 10.92% of the sender identities were good senders. By changing the minimum good reputation to 0.7, 51 (0.3%) additional sender identities were considered bad.

Graph 5 also validates this by showing the distribution of the number of sender identities vs. their lifetime. The number of sender identities with lifetime of 1 was about 8,000. However, as the lifetime increased, the number of sender identities became smaller and evenly distributed.

To prove our hypothesis that if the lifetime of the sender identity is long, the probability of it being a good identity is high, we plot the daily percentage of good and bad sender identities plotted against lifetime in Graph 6. The graph shows that the percentage of bad identities decreases as the lifetime increases, whereas the percentage increases for legitimate sender identity. Graph 7 validates this claim and shows the average reputations for all sender identities with same lifetimes. The curve for the average reputation for all identities shows a similar trend as percentage of good identities in Graph 6. Additionally, using a minimum good reputation of 0.5, credible sender identities had an average lifetime of 61.9 days while spammers had 17.47 days.



Graph 11: RepuScore: Reputation of the free email provider computed using two receiver organizations. Receiver organization 2 was introduced from day 107 of receiver organization 1.



Graph 12: Reputation of the free email provider using Volume-Enhanced RepuScore. Using volume, the global score is closer to the perceived reputation from both the receiver organizations.

Graph 8 shows the number of sender identities for which the reputation changed from being good to bad or vice versa. About 1.75% changed from being good to bad or vice versa corresponding to about 291 sender identities. There were only 8 sender identities whose reputation kept changing from good to bad or vice versa more than 15 times as their reputations hovered around 0.5.

4.3. Volume-Enhanced RepuScore

Graph 9 shows the reputation for a sender identity, corresponding to a popular free email service on the Internet. The sender identity had alternatively sent high and low spam rate to a single organization. Graph 10 shows the corresponding email volume and spam volume. We use the logs from the receiver organization 2 and reputation information from the receiver organization 1. Graph 9 shows the benefit of using volume enhancement as the sender identity reputation was varied with the email volume. From the graphs, we notice that the rate at which the reputation decreases for changes based on the email volume. For example, from interval 1 to 11, the volume-enhanced reputation is lower than the original reputation. At intervals 11 to 13, the reputation computed by volume-enhancement was higher than the original RepuScore algorithm. We note that with the help of Volume-Enhanced RepuScore, the slope of the reputation follows the email volume. The average reputation over 175 days for the sender was 0.82 using the original RepuScore and was about 0.662 using volume-enhanced RepuScore. On the average over 175 days, considering a minimum good reputation of 0.5, the sender was credible.

4.4. Combining Reputations from Two Receiver Organizations

We consider the effect of combining global reputation computed from two receiver organizations. For the sender identity discussed in Section 4.3, the receiver organization 2 transmitted about 38,100 authenticated emails of which 61 were spam in a span of 55 days. The receiver organization 2 started the evaluation of RepuScore from the reputation interval 107.

Graph 11 and Graph 12 shows results using RepuScore and Volume-Enhanced RepuScore. Our experiments show the accuracy of the reputation depends on the number of honest receiver organizations that start contributing to RepuScore. As the number of receiver organizations increase, the global reputation will be a weighted average of the reputation seen from different domains. In our example, as both receiver organizations did not maintain reputation for the other, the global score was a simple average.

5. Conclusion

Our previous work, RepuScore, is a collaborative reputation framework that calculates global reputation for sender identities by collecting reputation-views from multiple receivers. However, during our deployment, we noticed that the daily change in the email volume affected the reputation when only the spam-rate is used to calculate the score.

In this paper, we proposed Volume-Enhanced RepuScore that incorporate email volume in computing reputation in addition to the spam rate of the sender identity. By incorporating the email volume, a sender's reputation changed proportionately to the email volume.

We designed and developed a RepuScore plug-in for SpamAssassin to collect information about each email from mail servers. Using the plug-in, we deployed RepuScore at two organizations since 10/9/2007 and computed reputations for over 16,509 authenticated sender identities.

Our results show some interesting observations: a) identities with low reputation have a shorter lifetime compared to ones with high reputations; b) RepuScore was able to classify emails from about 42% of the authenticated sender identities corresponding to about 72% of the authenticated email volume; c) about 97.8% of the sender identities had reputation either near 0 or near 1. d) Average lifetime of good and bad sender identity was 61.9 and 17.47 days respectively as a large number of sender identities are created constantly that sent email only in one interval.

We invite further deployment of RepuScore framework. Please visit the website at: http://isr.uncc.edu/repuscore.

References

[1] E. Allman. DomainKeys Identified Mail (DKIM): Introduction and Overview, 2005.

[2] Alperovitch Dmitri, Judge Paul, Krasser Sven. Taxonomy of Email Reputation Systems, Distributed Computing Systems Workshops, 2007. ICDCSW 07.

[3] S. Biswas, R. Morris, "ExOR: Opportunistic Multi-Hop Routing for Wireless Networks", in the proceeding of ACM SIGCOMM '05, Philadephia, USA, 2005.

[4] Bradley Taylor. Sender Reputation in a Large Webmail Service, Third Conference on Email and Anti-Spam 2006.

[5] Certified Email, Goodmail Systems. www.goodmailsystems.com/certifiedmail.

[6] Chirita, P., Nejdl, W., Schlosser, M., Scurtu, O.: Personalized reputation management in P2P networks. Technical report, University of Hannover (2004)

[7] CipherTrust. TrustedSource: The Next-Generation Reputation System. White Paper. 2006.

[8] Dewan, P.; Dasgupta, P. Pride: peer-to-peer reputation infrastructure for decentralized environments. In Proceedings of the 13th international World Wide Web Conference (NY, USA, May 19 - 21, 2004).

[9] J.R. Douceur. The Sybil Attack. In Revised Papers From the First international Workshop on Peer-To-Peer Systems (March 07 - 08, 2002). P. Druschel, M. F. Kaashoek, and A. I. Rowstron, Eds. Lecture Notes In Computer Science, vol. 2429. Springer-Verlag, London.

[10] Goodrich M.T., Tamassia R., Yao D. Accredited DomainKeys: A Service Architecture for Improved Email Validation. CEAS 2005

[11] Habeas. Habeas SenderIndex. www.habeas.com/en-US/Receivers/SenderIndex/

[12] Habeas. Habeas SenderIndex. www.habeas.com/en-US/Receivers/SenderIndex/

[13] Hans Peter Brondmo, Margaret Olson, Paul Boissonneault. Project Lumos: A Solutions Blueprint for Solving the Spam Problem by Establishing Volume Email Sender Accountability, 2003.

[14] IronPort Study on Email Authentication Reveals Significant Adoption, 2006.

[15] Kamvar S. D.; Schlosser M. T.; Garcia-Molina H.. The EigenTrust Algorithm for Reputation Management in P2P Networks. In Proceedings of the Twelfth International World Wide Web Conference.

[16] McMillan R. Consumers to Lose $2.8 Billion to Phishers in 2006.

[17] Microsoft Corporation. Sender Id Framework – Executive Overview, 2004.

[18] Out-Law News. Over 90% of email is spam, says Spamhaus founder, 2006.

[19] Papaioannou, T. G. Stamoulis, G. D. 2004. Effective use of reputation in peer-to-peer environments. In Proceedings of the 2004 IEEE international Symposium on Cluster Computing and the Grid (April 19 - 22, 2004). CCGRID. IEEE Computer Society, Washington, DC, 259-268.

[20] Patrick Peterson, SIDF and DKIM overview Scorecard, Authentication Summit II, 2006.

[21] Prakash, V. V.; O'Donnell, A. Fighting spam with reputation systems. Queue 3, 9 (Nov. 2005), 36-41.

[22] Sender Score Email Reputation Management, Return Path. www.returnpath.com/delivery/senderscore.

[23] Sender Score Certified, Return Path Management. www.senderscorecertified.com.

[24] Singaraju G.; Kang B. RepuScore: Collaborative Reputation Management Framework for Email Infrastructure, USENIX 21th Large Installation System Administration Conference (LISA-2007).

[25] Shmatikov V.; Talcott C.. Reputation-based trust management. In Workshop on Issues in the Theory of Security (WITS), 2003.

[26] Srivatsa M.; Xiong L.; Liu L. TrustGuard: Countering Vulnerabilities in Reputation Management for Decentralized Networks. In 14th World Wide Web Conference (WWW 2005), Japan, 2005.

[27] Swamynathan, G.; Zhao, B. Y.; Almeroth, K. C. Exploring the feasibility of proactive reputations: Research Articles. Concurr. Comput.: Pract. Exper. 20, 2 (Feb. 2008).

[28] Wong M. W. Sender Authentication: what to do, Technical Document, 2004.

[29] Yahoo Inc. DomainKeys: Proving and Protecting Email Sender Identity.

[30] Yu B.; SinghM.P. An Evidential Model of Distributed Reputation Management. Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS). 2002.

[31] Yu B.; Singh M.P. Detecting Deception in Reputation Management. Proceedings of the 2nd International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Melbourne, ACM Press, 2003.

[32] Yu H.; Kaminsky M.; Gibbons P. B.; Flaxman A. D.. Defending against Sybil attacks via social networks. Proceedings of ACM SIGCOMM Conference, 2006.