Lightweight Client-side Methods for Detecting Email Forgery

Eric Lin[1], John Aycock[1*], and Mohammad Mannan[2]

[1] Department of Computer Science, University of Calgary,
2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4
{linyc,aycock}@ucalgary.ca
[2] Concordia Institute for Information Systems Engineering,
Faculty of Engineering and Computer Science, Concordia University,
1515 Ste-Catherine Street West, EV7 640, Montreal, Quebec, Canada H3G 2W1
mmannan@ciise.concordia.ca

Abstract. We examine a related, but distinct, problem to spam detection. Instead of trying to decide if email is spam or ham, we try to determine if email purporting to be from a known correspondent actually comes from that person – this may be seen as a way to address a class of targeted email attacks. We propose two methods, geolocation and stylometry analysis. The efficacy of geolocation was evaluated using over 73,000 emails collected from real users; stylometry, for comparison with related work from the area of computer forensics, was evaluated using selections from the Enron corpus. Both methods show promise for addressing the problem, and are complementary to existing anti-spam techniques. Neither requires global changes to email infrastructure, and both are done on the email client side, a practical means to empower end users with respect to security. Furthermore, both methods are lightweight in the sense that they leverage existing information and software in new ways, instead of needing massive deployments of untried applications.

1 Introduction

It is safe to say that very few people have friends and family who legitimately contact them about penis enlargement, Canadian pharmaceuticals, and lonely Russian women. However, email `From:` lines are trivial to forge, and passwords to user email accounts and social networking sites are easy to phish; usernames and accounts belonging to legitimate users may be co-opted to send spam.

More insidious is the threat of targeted attacks via email, which have gone up many-fold in the last few years; one report [23] noted these attacks have increased from 1–2 attacks/week in 2005 to 77 attacks/day in 2010. The same report stated that 6.3% of all phishing emails blocked in 2010 were targeted (spear) phishing attacks. In these attacks, an attacker may customize the email content for the target user, and/or masquerade as someone the target knows [8, 14]. Targeted phishing emails are more effective than bulk spam; e.g., one experiment

[*] Corresponding author

reported [18] significant differences in success rates of socially targeted phishing attacks (72%) vs. regular phishing attacks (16%).

Victims of these attacks come from all walks of life, including everyday users (e.g., friends/contacts of a target are asked to send money to help the target in need, stuck in a foreign country [8, 14]); government and defense executives (e.g., malicious attachments were received by a U.S. defense contractor purporting to be from a legitimate Pentagon sender [6]); and IT security professionals (e.g., targeted employees of RSA opened a malicious Excel attachment, entitled "2011 Recruitment Plan", allowing attackers to compromise sensitive information on widely used SecurID tokens [30]). Although very simple, these attacks have already caused significant financial damages, as well as increased risks to national security. From an attacker's point of view, these attacks are much less detectable, and estimated to yield much higher return than traditional spam (e.g., see [9]).

How can such targeted emails from seemingly legitimate sources be detected? We propose the use of IP geolocation (of known email-sending users and their email servers) and text stylometry analysis (of email content from known senders) to empower email recipients in detecting email forgery. The basic idea follows from familiar real-world experiences. For example, if a resident in the USA receives postal mail with a stamp from Nigeria, they are likely to be suspicious about the content of the mail, if it claims to be coming from the the IRS (Internal Revenue Service). In the same manner, when users receive a telephone call (assume a spoofed phone number) from someone purporting to be a friend/relative, users may detect the spoof from the differences in voice and language styles. We explore whether such "common sense" approaches can be used for email forgery detection.

The goal here is to provide more information to an end user for helping them decide whether the sender appears authentic or not. This information may be presented to the user by their MUA[3] based on classification done by the MUA itself or in the user's MTA or MDA; no global changes need be enacted to support our techniques. In fact, we have developed a client-side plug-in for the Thunderbird MUA that currently implements our IP geolocation method. However, we must determine the viability of anti-forgery techniques in real-world settings – the problem we address in this paper.

Looking at anti-spam techniques, it is easy to pick apart any new proposed method by attacking the assumptions that underlie it.[4] *All anti-spam technologies operate based on assumptions about spammer behavior. None of them are perfect and we do not claim that our forgery detection methods are either.* We are more concerned with having a wide range of anti-forgery methods so that defense in depth can be employed.

In terms of the threat model, we try to detect email forgery from two sources. First, compromised email accounts, where an attacker has access to a legitimate email account belonging to someone known to the target user. We observe that methods like SPF [32] and DKIM [1] do not help in this case. Second, arbitrary

[3] MUA = mail user agent; MTA = mail transfer agent; MDA = mail delivery agent.

[4] For a categorization of anti-spam techniques by assumption, see [16].

mail transmissions, where an attacker has forged the From: header to appear as if it comes from a sender known to the target user. We assume that attackers do not control the target's or the sender's email client or software platform. Common spam is not our focus. We also do not address phishing/scam emails from *unknown* senders (e.g., typical advance-fee frauds).

Our contributions may be summarized as follows. First, we have proposed and evaluated two methods for email forgery detection. Second, these methods purposely leverage existing information – in email headers, on the Internet – and existing, tested software, making our methods lightweight in terms of their requirements and implementation. Third, we tested our methods with real email. Getting access to spam samples is trivial, of course, but experiments like ours that require nonspam (i.e., ham) samples are much trickier, and demands that research be conducted observing ethical constraints (and sometimes the inability to capture certain data) to ensure privacy. Fourth, our methods are readily-deployable on the client side with no changes to email infrastructure needed.

The remainder of this paper looks at the classification techniques we have studied, with geolocation in Section 2 and stylometry in Section 3. Section 4 discusses related work, and Section 5 concludes.

2 Geolocation

The intuition behind geolocation for forgery detection is that most correspondents will send email from only a small number of physical locations. For example, an email arriving from China, claiming to be from a person who has only ever mailed from Canada previously, should be treated with a great degree of suspicion. This intuition will not always hold true, of course, and we revisit that and other limitations in the discussion below.

2.1 Experimental Design

To test our intuition, we performed a study on IP address information we found in saved email headers;[5] IP addresses can be mapped into a geographic location, and their stability may also provide a similar means to detect forgery, even if the IP addresses are not converted into a geolocation. It is important to note that we are not interested in the accuracy of geolocation or IP addresses. The purpose of our study is to find the *consistency* of senders' IP addresses and geolocation.

A script we wrote for data collection, run in study participants' accounts, looked for mailbox files in both mbox and dbx format. The script automatically extracted each sender's email address, domain, recipients, and IP address from a participant's saved email. Due to privacy issues, we only collected the (MD5) hash of a sender's mail address, the hash of a sender's domain, and the hash of a recipient's email address. None of the email body was read.

[5] Ethics approval for data collection (and restrictions on data collection) granted by University of Calgary Conjoint Faculties Research Ethics Board, file #6515.

Specifically, we were looking at the last `Received: from` header in each message (i.e., the first one added). We refer to these as *RF headers*. There are some problems with the IP addresses in the RF headers such as

1. The last RF header does not always contain an external (public) IP address. In this case, if the last RF header's IP address is internal, then we find the second to the last and so on, until the first external IP is found. That IP is taken as the sender's IP address.
2. Some mails are sent internally, so there are no external IP addresses at all. We discard all emails that only have internal IP addresses because internal IP addresses cannot be used to geolocate senders.

In both cases, we use the list of private IP address blocks from RFC 1918 [27].

We assume that most people don't save spam (anti-spam researchers not being "most people"), so we assume data collected is mostly from legitimate senders. We do, however, filter out common spam folders in the study, like "Spam" and "Junk."

A total of ten subjects participated in our study, selected via convenience sampling. After filtering out the emails without an external IP address, a total number of 73,652 emails were collected. Out of 73,652 emails, 6909 unique senders were extracted and 2838 unique domains were found.

2.2 Analysis

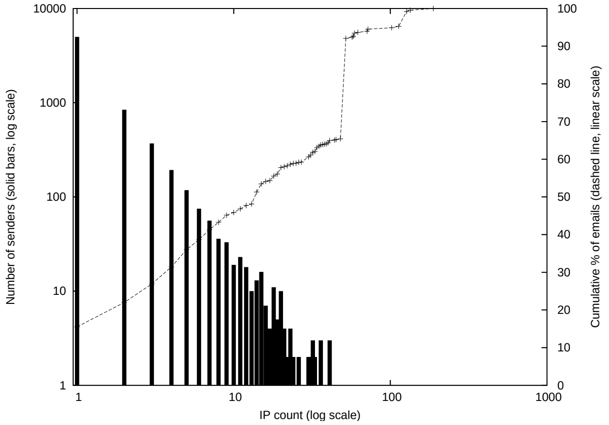


Fig. 1. IP address counts vs. number of senders and amount of email sent.

Looking at the number of unique IP addresses a sender has shows a large range (Figure 1). 72% of senders have only one IP address, 85% of senders have one or two IP addresses, and 15% of senders have three or more IP addresses.

However, these results are tempered by the number of emails that are sent from different numbers of IP addresses. Only 16% of emails were sent by senders with one IP address, 22% of emails were sent by senders with one or two IP addresses, and the majority, 78%, of emails were sent by senders with more than three IP addresses.

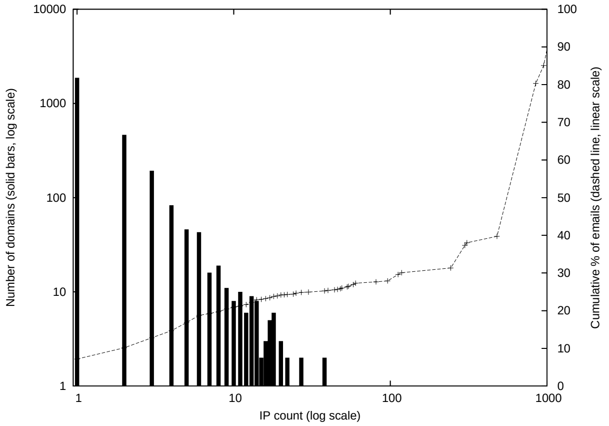


Fig. 2. IP address counts vs. number of domains and amount of email sent.

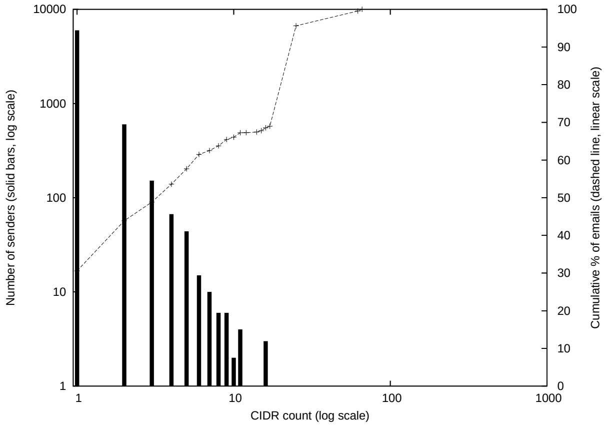


Fig. 3. CIDR address counts vs. number of senders and amount of email sent.

A sender's email address also relates to a domain, which is the email provider's name that the email address is hosted at. Figure 2 shows an even larger variance of IP addresses amongst the domains. 66% of domains have only one IP address,

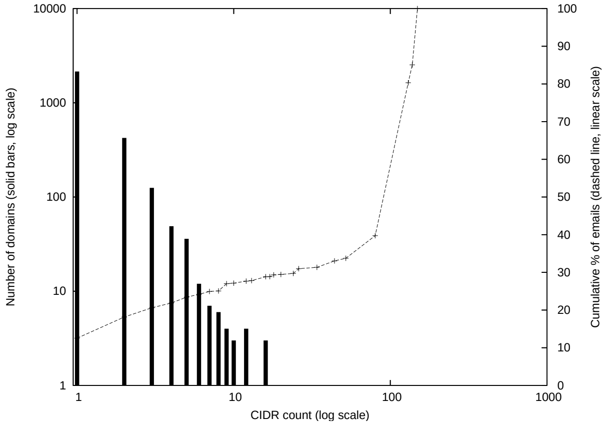


Fig. 4. CIDR address counts vs. number of domains and amount of email sent.

82% have one or two IP addresses, and the remaining 18% domains have three or more IP addresses. Furthermore, there is one domain (not shown) that has 1128 different IP addresses. We observed most emails were sent from domains with more than three IP addresses. That is, 90% of emails were sent from domains which have more than three IP addresses, and the remaining 10% of emails were sent from domains which have one or two IP addresses.

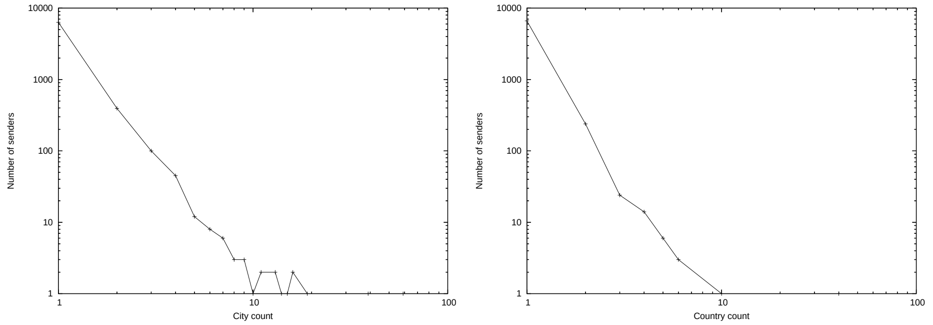
One factor that might cause large sets of IP addresses for some senders is dynamic IP addresses. In an attempt to mitigate this effect, we used `whois` to map senders' IP addresses into their corresponding CIDR address. With CIDR lookups, the IPs for a single sender reduced significantly (Figure 3). That is, 87% of senders have one CIDR address and 95% of senders have one or two CIDR addresses, and only 5% of senders have three or more CIDR addresses. In terms of number of emails, we observed an increase in the emails sent from senders with one or two CIDR addresses to 44%, and a decrease in emails sent from senders with three or more CIDR addresses to 56%.

We performed the same analysis on the domains' CIDR addresses. Figure 4 shows that the number of CIDR addresses of domains also reduced significantly. 76% of domains have only one CIDR address, 91% have one or two CIDR addresses, and the remaining 9% of domains have three or more CIDR addresses. After mapping IP addresses to CIDR addresses, there is a significant difference in the total number of emails sent from different groups of CIDR addresses: the 91% of domains with one or two CIDR addresses contribute 18% of total emails.

Turning now to geolocation, we mapped senders' IP addresses into city and country granularity using `IPInfoDB`.[6] Again, we are interested in consistency rather than mapping accuracy.

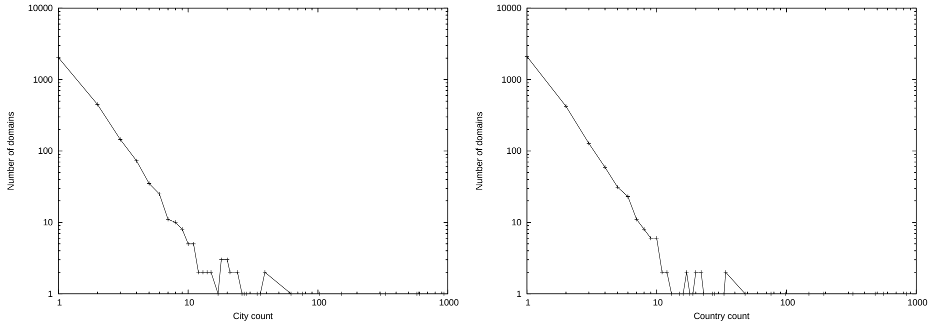
A total number of 1310 unique cities and 113 unique countries were found in our data. From Figure 5a we observed 92% of senders were only located in one

[6] `http://ipinfodb.com`



(a) (b)

Fig. 5. Number of senders vs. city (a) and country (b) counts, log-log scale.



(a) (b)

Fig. 6. Number of domains vs. city (a) and country (b) counts, log-log scale.

city, over 97% senders were located in one or two cities, and just 3% of senders were located in three or more cities.

Figure 5b shows an even stronger correlation between senders and their countries. 96% of senders only send email from one country, 99% of senders were located in one or two countries, and only 1% of senders were located in three or more countries.

We find there is a strong correlation between senders and their city/country location, but this is less so for the domains. In Figure 6a we observed 72% of domains located in one city, 88% domains located in one or two cities, and 12% of domains located in three or more cities. Figure 6b shows 74% of domains send email from one country, 89% of domains are located in one or two countries, and the remaining 11% of domains are located in three or more countries.

2.3 Discussion

Our results indicate that the majority of senders have four properties. They have a small set of IP addresses; they have a small set of CIDR addresses; they have highly consistent city geolocation; they have highly consistent country geolocation. From any or all of these, we may construct an email forgery classifier that will flag emails from known senders whose email comes from an unexpected location. The fact that many emails, in terms of volume, do not fit this profile is not a major concern, as we are not trying to construct a general anti-spam solution, but detect forgery for known senders – the important criterion is that the vast majority of senders *do* have these properties.

Domain information was less stable and, for example, we observed a large number of emails were sent from domains with more than three IP and CIDR addresses. This might indicate large web-based mail providers (e.g., Yahoo!, Gmail, Hotmail), or may be attributable to some participants retaining some spam in their saved mail.

There are several limitations with our data analysis. First, while we collected a large number of real emails, the sample may be biased, and it would be helpful to repeat this experiment with a larger sample size. Demographic information about the study participants would be interesting to collect in a larger study, although the demographic of email senders may be equally interesting (but much harder to collect). Second, emails which are sent from some web-based email clients (e.g., Gmail) do not reveal their true IP addresses. Often, the only traceable IP is the the email provider's mail server IPs, and the sender's true IP address is not shown in the RF header. If a forged email is sent from the same web-based mail provider as a legitimate email, then it is not possible to distinguish them by IP addresses. In order to solve this problem, we recommend that web-based email providers log the sender's IP from their TCP connection into the email header; this can be used to detect forged emails sent from web-based providers. Third, geolocation databases will inevitably have some minor inaccuracies that may affect city and country mapping. If more accuracy is needed, it is possible to query multiple geolocation databases. Finally, some senders might never have a consistent geolocation. That is, it is possible that a person travels consistently and sends email from different geolocations, or a person always uses anonymous proxies (e.g., Tor [12]). For this type of user, no geolocation-based technique will apply. While we know our method is not a perfect solution that is suitable for all senders, it can be used for a majority of senders, and it can be combined with other forgery detection mechanisms, like stylometry.

3 Stylometry

The stylometry problem may be stated informally as follows. Say that Alice receives a piece of email that purports to be from Bob. Alice and Bob have exchanged emails in the past, and consequently Alice has a collection of past email messages from Bob. Can the characteristics of Bob's past email be used to determine if the latest email is really from Bob? Any solution, as part of an email system, must be fully automated and not require preselection of features, have low overhead, make a decision using short inputs, and be scalable from one past email correspondent to arbitrarily many. Ideally, a solution should not be limited to English text; obviously, a solution should have high accuracy.

It seemed to be a straightforward application of existing computer forensic techniques, at first. However, upon closer examination, most previous work on authorship analysis addresses a different problem: let $\{S_1, S_2...., S_n\}$, $n \geq 2$, be the set of senders in the training database. There is a collection of email E_i for each $S_i \in \{S_1, S_2...., S_n\}$, and $|E_i| \geq 1$. Upon receiving a suspect email, M, with unknown authorship, identify the author S_a from $\{S_1, S_2...., S_n\}$ whose writing style most closely matches the writing style of M.

In other words, the "usual suspects" have already been rounded up; it is simply a question of "whodunnit?" This unfortunately does not work in our case. Our solution has to work even if there is only one sender present in the training database, and there is no guarantee that the sender comes from a known set. Much seemingly related work is thus not applicable.

Instead, our problem falls into a subarea of authorship analysis called "author identification" in Zheng et al.'s taxonomy, which computes 'the likelihood of a particular author having written a piece of work by examining other works produced by that author' [34, page 60]. Formally, let $\{S_1, S_2...., S_n\}$, $n \geq 1$, be the set of senders in the training database. There is a collection of email E_i for each $S_i \in \{S_1, S_2...., S_n\}$, and $|E_i| \geq 1$. Upon receiving a suspect email, M, which claims it is from author S_a, where $S_a \in \{S_1, S_2...., S_n\}$, identify how likely it is that M is written by S_a.

3.1 Our Method

We based our stylometry system on the SpamBayes statistical spam filter[7] [24] inheriting, among other things, various canonicalizations of emails and a χ^2 method of combining the probability scores for individual tokens in a new email.

[7] http://spambayes.sourceforge.net, version 1.0.4.

We switched the tokenizer to an N-gram tokenizer (N = 5), because we found it gave us better accuracy in our experiments (similar to Kanaris et al. [20]) and also allows our system to be language-independent.

The usual approach would be to train SpamBayes with ham and spam, and create a single database. Instead, we had a separate database for each known email sender, training on legitimate email from each respective sender and telling SpamBayes that the email was all ham.

As a consequence of training on ham from a given sender only, the assumption that the training corpora are half ham and half spam no longer applies. Therefore, when classifying a new email, we have moved away from Robinson's calculation [28] and instead compute a token's probability score using the simplified formula

$$prob = \frac{P \times S + n}{S + n} \tag{1}$$

Here, n = number of times the token occurs in the dictionary; P is the probability of an unknown token. From testing on the Enron corpus, $P = 0.3$ yields the best result. S is the strength of an unknown token. It adjusts the weight of probability of this token by counting n. $S = 0$ means the token is always believed 100%; from testing of the Enron corpus, $S = 0.6$ yields the best result.

3.2 Experiments

The Enron corpus [19] is the data source used in our experiment; it contains 200,399 real emails from 158 users.

Forwarded emails written by other people can affect author identification results, so they must be removed. The Enron corpus contains many different formats for forwarding information in emails, so we manually filtered out forwarded content.[8] Manual selection might bias the result, but we tried to minimize the impact: we only filtered out forwarded contents of each email, and other than stripping out forwarded content, the original text remained unchanged.

Each experiment was run 30 times and the arithmetic mean is reported here. Each experiment included a training and testing phase. Training and testing emails were selected randomly for each test to reduce bias.

We first trained the engine with a set of legitimate emails from a sender, S. Emails from other senders were taken as the testing emails. The `From:` header from all testing emails was forged to the email address of the sender S, then each testing email was sent to the engine to query for the probability of the email being written by the sender S.

We use the matrix in Table 1 to calculate statistical measurement in this experiment. True negative (TN) measurement, which means the engine classifying a forged email as not legitimate, is the most important measurement because it represents whether or not our method successfully classifies a forged email, so

[8] We note that Iqbal et al. [17] also manually filtered Enron emails in their study. A production system would, of course, do this automatically, but we wanted to ensure a clean data set for experiments.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Table 1. Confusion matrix

our precision and recall are focused on it rather than true positives; F_1 is the normalized mean of precision and recall.

$$Recall\ (R) = \frac{TN}{TN + FP} \tag{2}$$

$$Precision\ (P) = \frac{TN}{TN + FN} \tag{3}$$

$$F_1 = \frac{2PR}{P + R} \tag{4}$$

The probability of classifying a forged email with our method is further broken into five intervals for detailed analysis: [0, 0.25), [0.25,0.5), 0.5, (0.5, 0.75), and [0.75, 1]. A probability of 0 means the email is most likely not composed by the author, 1 means it most likely is, and 0.5 means not sure. Any value below 0.5 implies TN, 0.5 implies neutral, and greater than 0.5 implies FP.

We ran this experiment with four different sets of training emails: 10, 20, 30, and 40. Testing against different numbers of training emails is important because of two things that occur in real world conditions. First, it is not necessarily the case that users always train a large fraction of total emails across all senders – some senders might have more training emails than the others, and they will not be perfectly balanced. Second, users might only train the engine with a fixed number of training emails from a sender and use the fixed set of training emails for all further classification.

We randomly selected six authors for this experiment. For each author, we first trained the engine with a fixed number of emails (randomly selected), then we took all emails from all other authors (5 authors, 50 emails each, for a total of 250 emails). These 250 testing emails' `From:` header email address was forged to the email address of the author trained in the engine. Then each testing email was sent to the engine for querying the probability score.

We show the detailed classification scores of 10 training emails divided into intervals in Table 2; Table 3 shows the corresponding precision, recall, and F_1 values. Our method was almost perfect on recall, precision, and F_1 against forged emails with only 10 training emails from each author. Table 4 gives the F_1 values computed from the tests of all different numbers of training emails (e.g. 10, 20, 30, and 40). In general, our method is still able to produce close to 100% F_1 values against forged emails.

Of course, forged emails will (hopefully) be the exception rather than the norm, and we also looked at the classification accuracy for the sender's own

	Authors					
	Allen	Beck	Blair	Cash	Haedicke	Heard
[0, 0.25)	0	0	0	0	0	0
[0.25,0.5)	6942	7292	7112	7100	7124	7037
0.5	528	152	315	352	345	405
(0.5, 0.75)	0	56	43	18	1	28
[0.75, 1]	0	0	0	0	0	0

Table 2. Forging emails: classification scores divided into intervals of forging email senders, training 10 emails.

	Authors					
	Allen	Beck	Blair	Cash	Haedicke	Heard
R	100%	99.0%	99.4%	99.7%	100%	99.6%
P	100%	100%	100%	100%	100%	100%
F_1	100%	99.6%	99.7%	99.9%	99.9%	99.8%

Table 3. Forging emails: precision, recall, and F_1 of forging email senders, training 10 emails.

	Authors					
	Allen	Beck	Blair	Cash	Haedicke	Heard
F_1 (T=10)	100.00%	99.60%	99.70%	99.90%	99.90%	99.80%
F_1 (T=20)	99.80%	98.00%	98.60%	99.10%	99.40%	98.80%
F_1 (T=30)	99.30%	95.60%	97.60%	97.50%	98.90%	97.40%
F_1 (T=40)	99.00%	92.70%	96.40%	94.90%	97.70%	96.00%

Table 4. Forging emails: F_1 values of different number of training emails.

emails. For each sender, we took 14 emails of the sender and trained the engine, and then the remaining emails from the sender were used for testing emails. Table 5 shows the results of testing a sender's own emails. Our method is able to perform adequately in terms of precision, recall, and F_1.

	Authors					
	Allen	Beck	Blair	Cash	Haedicke	Heard
R	100%	100%	100%	100%	100%	100%
P	62%	82%	91%	84%	65%	98%
F_1	77%	90%	95%	91%	79%	99%

Table 5. Precision, recall, and F_1 of each sender's own emails: 14 training emails.

3.3 Discussion

Our method is able to classify both legitimate and forged emails with good classification accuracy. As with geolocation, the classification need not be perfect, as it can be used in conjunction with other methods.

We assume that an attacker does not know about the writing style of a sender; thus, one limitation of our method is that it might not work as well when the attacker gains knowledge about the sender. A possible scenario: what if a sender's computer is compromised and becomes a zombie machine, giving the attacker access to emails saved on the machine? The attacker can make use of the saved emails and forge emails with a very similar writing style to the legitimate sender; this type of attack was discussed in [3]. Manual attempts to imitate writing style are examined with respect to authorship analysis in [5]. In the case of automated or manual style imitation, our method might not be able to distinguish between legitimate and forged emails because the writing styles would be very similar.

4 Related Work

Related geolocation work. There is much existing work that examines the characteristics of spam, spammers, and IP addresses.

Ramachandran et al. [26] studied spammers' traffic at the network level. Among other things, they found that both spam and legitimate emails can come from similar IP address spaces, so it is impossible to distinguish spam and non-spam email by IP addresses alone, in general. However, they did not examine legitimate emails in depth, and did not examine the role played by persistence in legitimate senders' IP addresses.

Cook et al. [10] looked to see if there were any indicators that spam was about to be sent from an IP address, but did not find any strong signs that conclusively flagged forthcoming spam. Again, this work is not looking at the same problem that we are.

Gomes et al. [15] collected 360,000 emails sent to a university and used SpamAssassin to classify the emails into spam and ham, gathering data about both. Most relevant is their observation that 'on average, a single sender domain is associated with 15 different IP addresses, whereas the average number of different domains per sender IP address is only 6' (page 359). However, this is only an aside and they do not discuss or analyze this further.

Sanchez et al. [29] want to find out whether or not spammers lie about their IP addresses in `Received:` headers. Their results showed this was a rarity, which is good news for forgery detection based on this information.

Xie et al. [33] studied dynamic IP addresses, discovering 'IP-to-host bindings changing from several hours to several days' (page 302). This suggests that the CIDR or city/country geolocation may be preferable for anti-forgery than the IP address itself.

The situation does not improve for emails sent from mobile phones. Balakrishnan et al. [4] studied geolocation for mobile phones' IP addresses, finding it wildly inaccurate at times. While they concluded that geolocation was 'impossible,' we observe that this does not mean that the accuracy will not improve over time, and in any case we only require consistency.

Email IP address geolocation falls into the subproblem of IP geolocation in Muir et al.'s [25] classification because, in our case, we will have an IP address but not necessarily an active TCP connection. Although IP address geolocation has many limitations as Muir et al. discussed, again we are interested in consistency rather than mapping accuracy.

Related stylometry work. A wide variety of techniques have been brought to bear on stylometry. Calix et al. [7] used a K-nearest neighbor (KNN) algorithm along with 55 stylistic features to classify the author of unknown emails. Corney [11] used a support vector machine (SVM) technique with extensive experiments to determine good features. Iqbal et al. [17] used features combined to create what they called 'write-prints,' meant to be analogous to fingerprints. Argamon et al. [2] used an exponentiated gradient learning algorithm. Frantzeskou et al. [13] used an N-gram tokenizer for analyzing source code authorship, along with a similarity measure of their own devising. A χ^2 approach was used by Vogel and Lynch [31] for classical works, and also by Luyckx and Daelemans [22]. The work we have found in this area typically suffers from one or more difficulties with respect to our problem criteria. For example, manually selecting features rather than automatically discovering them; expecting a small set of "suspects" to choose from; needing large amounts of text; not being language-independent. In comparison, our solution based on SpamBayes is both simple and effective.

5 Conclusion and Future Work

This paper examined the detection of email forgeries, mails that claim to be from a known sender, which includes a class of targeted email attacks. We addressed the problem by using two characteristics of the legitimate sender's email – geolocation and stylometry. While neither proffers perfect detection, they can be used in combination with each other or existing anti-spam techniques. All analysis can be done on the targeted user's side, without requiring a global overhaul of email, and leverages information and software that exists today.

Besides adding stylometry into our client-side plug-in, future work will examine other sources of sender location data. For example, using a sender's location from Facebook or Foursquare, or a special mobile application that posts encrypted location data to Twitter, may yield useful data for validating senders.

Acknowledgments. The authors' research is supported by the Natural Sciences and Engineering Research Council of Canada via ISSNet, the Internetworked Systems Security Network. This paper is based on the first author's thesis [21].

References

1. E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, and M. Thomas. DomainKeys Identified Mail (DKIM) Signatures. RFC 4871 (Proposed Standard), May 2007. Updated by RFC 5672.
2. S. Argamon, M. Šarić, and S. S. Stein. Style mining of electronic messages for multiple authorship discrimination: first results. In *9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 475–480, 2003.
3. J. Aycock and N. Friess. Spam zombies from outer space. In *15th Annual EICAR Conference*, pages 164–179, 2006.
4. M. Balakrishnan, I. Mohomed, and V. Ramasubramanian. Where's that phone?: geolocating IP addresses on 3G networks. In *9th ACM SIGCOMM Conference on Internet Measurement*, pages 294–300, 2009.
5. M. Brennan and R. Greenstadt. Practical attacks against authorship recognition techniques. In *21st Innovative Applications of Artificial Intelligence Conference*, pages 60–65, 2009.
6. BusinessWeek. The new e-spionage threat. Cover story (Apr. 10, 2008). `http://www.businessweek.com/magazine/content/08_16/b4080032218430.htm`.
7. K. Calix, M. Connors, D. Levy, H. Manzar, G. McCabe, and S. Westcott. Stylometry for e-mail author identification and authentication. *Proceedings of CSIS Research Day, Pace University*, 2008.
8. CBC News. Ottawa man victim of Facebook, email scam. News article (Mar. 2, 2011). `http://www.cbc.ca/news/canada/ottawa/story/2011/03/02/ottawa-facebook-scam.html`.
9. Cisco.com. Email attacks: This time its personal. Online resource (June 2011). `http://www.cisco.com/en/US/prod/collateral/vpndevc/ps10128/ps10339/ps10354/targeted_attacks.pdf`.
10. D. Cook, J. Hartnett, K. Manderson, and J. Scanlan. Catching spam before it arrives: domain specific dynamic blacklists. In *2006 Australasian Workshops on Grid Computing and e-Research*, pages 193–202, 2006.
11. M. Corney. Analysing e-mail text authorship for forensic purposes. Master of Information Technology thesis, Queensland University of Technology, 2003.
12. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *13th USENIX Security Symposium*, pages 303–320, 2004.

13. G. Frantzeskou, E. Stamatatos, S. Gritzalis, and S. Katsikas. Source code author identification based on n-gram author profiles. In *IFIP International Federation for Information Processing*, pages 508–515, 2006.
14. D. F. Gallagher. E-mail scammers ask your friends for money. New York Times. Blog article (Nov. 9, 2007). http://bits.blogs.nytimes.com/2007/11/09/e-mail-scammers-ask-your-friends-for-money/.
15. L. H. Gomes, C. Cazita, J. M. Almeida, V. Almeida, and W. Meira, Jr. Characterizing a spam traffic. In *4th ACM SIGCOMM Conference on Internet Measurement*, pages 356–369, 2004.
16. R. Hemmingsen, J. Aycock, and M. Jacobson, Jr. Spam, phishing, and the looming challenge of big botnets. In *EU Spam Symposium*, 2007.
17. F. Iqbal, R. Hadjidj, B. C. Fung, and M. Debbabi. A novel approach of mining write-prints for authorship attribution in e-mail forensics. *Digital Investigation*, 5(Supplement 1):S42 – S51, 2008.
18. T. Jagatic, N. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Commun. ACM*, 50(10):94–100, 2007.
19. L. Kaelbling. Enron email dataset. CALO Project, http://www.cs.cmu.edu/~enron/, August 21 2009.
20. I. Kanaris, K. Kanaris, J. Houvardas, and E. Stamatatos. Words vs. character n-grams for anti-spam filtering. *Int. Journal on Artificial Intelligence Tools*, 2007.
21. E. Lin. Detecting email forgery. Master's thesis, University of Calgary, 2011.
22. K. Luyckx and W. Daelemans. Authorship attribution and verification with many authors and limited data. In *22nd International Conference on Computational Linguistics*, pages 513–520, 2008.
23. MessageLabs. MessageLabs intelligence: 2010 annual security report. http://www.messagelabs.com/mlireport/MessageLabsIntelligence_2010_Annual_Report_FINAL.pdf.
24. T. A. Meyer and B. Whateley. SpamBayes: Effective open-source, Bayesian based, email classification system. In *1st Conference on Email and Anti-Spam*, 2004.
25. J. A. Muir and P. C. Van Oorschot. Internet geolocation: Evasion and counterevasion. *ACM Comput. Surv.*, 42(1):1–23, 2009.
26. A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. *SIGCOMM Comput. Commun. Rev.*, 36(4):291–302, 2006.
27. Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918 (Best Current Practice), Feb. 1996.
28. G. Robinson. A statistical approach to the spam problem. *Linux Journal*, 107, Mar. 2003.
29. F. Sanchez, Z. Duan, and Y. Dong. Understanding forgery properties of spam delivery paths. In *Proc. 7th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference (CEAS)*, pages 13–14, July 2010.
30. ThreatPost.com. RSA: SecurID attack was phishing via an Excel spreadsheet. Blog article (Apr. 1, 2011). http://threatpost.com/en_us/blogs/rsa-securid-attack-was-phishing-excel-spreadsheet-040111.
31. C. Vogel and G. Lynch. Computational stylometry: Who's in a play? In *Verbal and Nonverbal Features of Human-Human and Human-Machine Interaction: COST Action 2102 International Conference, Revised Papers*, pages 169–186, 2008.
32. M. Wong and W. Schlitt. Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1. RFC 4408 (Experimental), Apr. 2006.
33. Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber. How dynamic are IP addresses? In *2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 301–312, 2007.

34. R. Zheng, Y. Qin, Z. Huang, and H. Chen. Authorship analysis in cybercrime investigation. In *1st NSF/NIJ Conference on Intelligence and Security Informatics*, pages 59–73, 2003.