Privacy-Aware Collaborative Spam Filtering

Kang Li, *Member*, *IEEE*, Zhenyu Zhong, *Student Member*, *IEEE*, and Lakshmish Ramaswamy, *Member*, *IEEE*

Abstract—While the concept of collaboration provides a natural defense against massive spam e-mails directed at large numbers of recipients, designing effective collaborative anti-spam systems raises several important research challenges. First and foremost, since e-mails may contain confidential information, any collaborative anti-spam approach has to guarantee strong privacy protection to the participating entities. Second, the continuously evolving nature of spam demands the collaborative techniques to be resilient to various kinds of camouflage attacks. Third, the collaboration has to be lightweight, efficient, and scalable. Toward addressing these challenges, this paper presents ALPACAS—a privacy-aware framework for collaborative spam filtering. In designing the ALPACAS framework, we make two unique contributions. The first is a feature-preserving message transformation technique that is highly resilient against the latest kinds of spam attacks. The second is a privacy-preserving protocol that provides enhanced privacy guarantees to the participating entities. Our experimental results conducted on a real e-mail data set shows that the proposed framework provides a 10 fold improvement in the false negative rate over the Bayesian-based Bogofilter when faced with one of the recent kinds of spam attacks. Further, the privacy breaches are extremely rare. This demonstrates the strong privacy protection provided by the ALPACAS system.

Index Terms—Distributed systems, collaboration, spam, privacy.

1 Introduction

Statistical filtering (especially Bayesian filtering) has long been a popular anti-spam approach, but spam continues to be a serious problem to the Internet society. Recent spam attacks expose strong challenges to the statistical filters, which highlights the need for a new anti-spam approach.

The economics of spam dictates that the spammer has to target several recipients with identical or similar e-mail messages. This makes collaborative spam filtering a natural defense paradigm, wherein a set of e-mail clients share their knowledge about recently received spam e-mails, providing a highly effective defense against a substantial fraction of spam attacks. Also, knowledge sharing can significantly alleviate the burdens of frequent training stand-alone spam filters.

However, any large-scale collaborative anti-spam approach is faced with a fundamental and important challenge, namely *ensuring the privacy of the e-mails among untrusted e-mail entities*. Different from the e-mail service providers such as Gmail or Yahoo mail, which utilizes spam or ham (non-spam) classifications from all its users to classify new messages, privacy is a major concern for cross-enterprise collaboration, especially in a large scale. The idea of collaboration implies that the participating users and e-mail servers have to share and exchange information about the e-mails (including the classification result). However, e-mails are generally considered as private communication between the senders and the recipients, and they often contain personal and confidential information. Therefore, users and organizations are not comfortable sharing information about their e-mails until and unless they are assured that no one else (human or machine) would become aware of the actual contents of their e-mails. This genuine concern for privacy has deterred users and organizations from participating in any large-scale collaborative spam filtering effort.

To protect e-mail privacy, digest approach has been proposed in the collaborative anti-spam systems to both provide encryption for the e-mail messages and obtain useful information (*fingerprint*) from spam e-mail. Ideally, the digest calculation has to be a one-way function such that it should be computationally hard to generate the corresponding e-mail message. It should embody the textual features of the e-mail message such that if two e-mails have similar syntactic structure, then their fingerprints should also be similar. A few distributed spam identification schemes, such as Distributed Checksum Clearinghouse (DCC) [2] and Vipul's Razor [3] have different ways to generate fingerprints. However, these systems are not sufficient to handle two security threats: 1) *Privacy breach* as discussed in detail in Section 2 and 2) *Camouflage attacks*, such as character replacement and good word appendant, make it hard to generate the same e-mail fingerprints for highly similar spam e-mails.

To simultaneously achieve the conflicting goals of ensuring the privacy of the participating entities and effectively and resiliently harnessing the power of collaboration for countering spam, we design a particular framework and name it "**A** **L**arge-scale **P**rivacy-**A**ware **C**ollaborative **A**nti-spam **S**ystem" (**ALPACAS**).

In designing the ALPACAS framework, this paper makes two unique contributions: 1) We present a resilient fingerprint generation technique called *feature-preserving transformation* that effectively captures the similarity information of the e-mails into their respective encodings, so that it is

- *K. Li and L. Ramaswamy are with the Department of Computer Science, University of Georgia, 415 Boyd GSRC, Athens, GA 30602. E-mail: {kangli, laks}@cs.uga.edu.*
- *Z. Zhong is with Secure Computing, 4800 N Point Pkwy, Alpharetta, GA 30022. E-mail: edward.zhenyu@gmail.com.*

Manuscript received 1 Feb. 2008; revised 15 July 2008; accepted 16 July 2008; published online 29 July 2008.
Recommended for acceptance by M. Singhal.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2008-02-0042.
Digital Object Identifier no. 10.1109/TPDS.2008.143.

possible to perform fast and accurate similarity comparisons without the actual contents of the e-mails. Further, this technique also ensures that it is computationally infeasible to reverse-engineer the contents of an e-mail from its encoding. 2) For further enforcing the privacy protection, a privacy-preserving protocol is designed to control the amount of information to be shared among the collaborating entities and the manner in which the sharing is done.

We evaluate the proposed mechanisms through series of experiments on a real e-mail corpus. The results demonstrate that the ALPACAS framework has a comparable overall filtering accuracy to the traditional stand-alone statistical filters. Furthermore, ALPACAS resists various kinds of spam attacks effectively. For good word attack, ALPACAS has 10 times better false negative rates than both DCC and Bogofilter [4], a well-known Bayesian-based spam filter. For character replacement attack, ALPACAS shows a 30 times better false negative rate than DCC and 9 times better false negative rate than Bogofilter. ALPACAS also provides strong privacy protection. The probability of a ham message to be guessed correctly by a remote collaborating peer is well controlled below 0.001.

2 Motivation and Prior Work

Researchers have proposed many spam resistance approaches including white and black lists [5], statistical filtering [6], network analysis [7], [8], and sender authentication [9]. A single commercial product often employs many of these approaches concurrently.

2.1 Limitations of Statistical Filtering Techniques

Statistical filtering is currently the predominant anti-spam approach. The central idea of all statistical filters is to assign each word (more generally *token*) with a spam likelihood value and a ham likelihood value and classify e-mails based on the likelihood values of the words appearing in them. Naive Bayesian classifier, which is a popular machine learning-based statistical filter, generates the spam and ham likelihood values of the tokens based on the statistics of their appearances in a set of training data. For each newly arriving message, this technique calculates a score based on the spam and ham likelihood values of its tokens, which is then used for classifying the message.

With significant amount of research efforts devoted to improving its accuracy, statistical filters have been reasonably successful in filtering traditional types of spam messages when they are trained with sufficient data. However, these stand-alone statistical filters suffer from two major limitations. First, statistical filters are highly vulnerable to a class of attacks that are intended to confuse them by appending ham-like material or reducing the spam words in the e-mails. For example, in the good word attack, the spammer appends large numbers of *good words* (those that appear mostly in ham messages) to the end of spam e-mails, thereby misleading the statistical filters to classify them as ham. Similarly, *Picospams* are extremely small e-mail messages, and they hardly contain any word that can be used by statistical filters for classification. Our experiments (see Section 4) show that the effectiveness of the Bayesian filter can deteriorate by a staggering 55 percent, when only 20 percent good words are appended to the e-mail.

Second, most statistical filters suffer from limited training set. Since the training sets are the basis upon which the spam and ham likelihood values are computed, the statistical filters are very sensitive to the accuracy and completeness of the training sets. While reasonable spam data sets are publicly available [10], [11], privacy concerns has deterred users and organizations from participating in any ham archiving efforts. Thus, the sizes of publicly available ham data sets are small fractions of their spam counterparts. Moreover, in order to deal with constantly evolving spam mechanisms, statistical techniques need continuous streams of ham and spam training sets, which are generally not available [12], [13], [14], [15], [16]. These factors have adversely affected the classification accuracies of statistical filters.

2.2 State of the Art in Collaborative Anti-Spam Systems

Prior efforts on coordinated real-time spam blocking include DCC [2], Vipul's Razor [3], SpamNet [17], P2P spam filtering [18], [19], and SpamWatch [20]. We discuss the drawbacks of the existing collaborative anti-spam schemes using DCC as a representative example.

The DCC system attempts to address the privacy issue by using hash functions. Here, the participating servers do not share the actual e-mails they have received and classified. Rather they share the e-mails' *digests*, which are computed through hashing functions such as MD5 over the e-mail body. When an e-mail arrives at a mail server, it queries the DCC system with the message digest. The DCC system replies back with the recent statistics about the digest (such as the number of instances of this digest being reported as spam). DCC suffers from two major drawbacks: First, since hashing schemes like MD5 generate completely different hash values even if the message is altered by a single byte, the DCC scheme is successful only if exactly the same e-mail is received at multiple collaborative servers. DCC develops fuzzy checksums to improve the robustness by selecting parts of the messages based on a predefined dictionary. However, spammers can get around this technique by attaching a few different words to each e-mail.

Second, the DCC scheme does not completely address the privacy issue. A closer examination reveals that the confidentiality of the e-mails can be compromised during the collaboration process of DCC. Thus, it violates the privacy requirement from the e-mail sender for maintaining the confidentiality of the recipients when he wants to deliver e-mails to multiple recipients by using "Bcc:." In particular, one DCC server can possibly infer who else receives the same e-mail by comparing the querying fuzzy checksum. Assuming DCC uses perfect hash function, consider the scenario wherein an e-mail server EA_i received a ham e-mail M_a. Suppose another e-mail server, say EA_j, receives an identical e-mail later and sends its fuzzy checksum to EA_i. Since EA_i had seen this e-mail before, it immediately discovers that EA_j has also received the same e-mail M_a. We refer to this type of privacy compromise as *inference-based privacy breaches*.

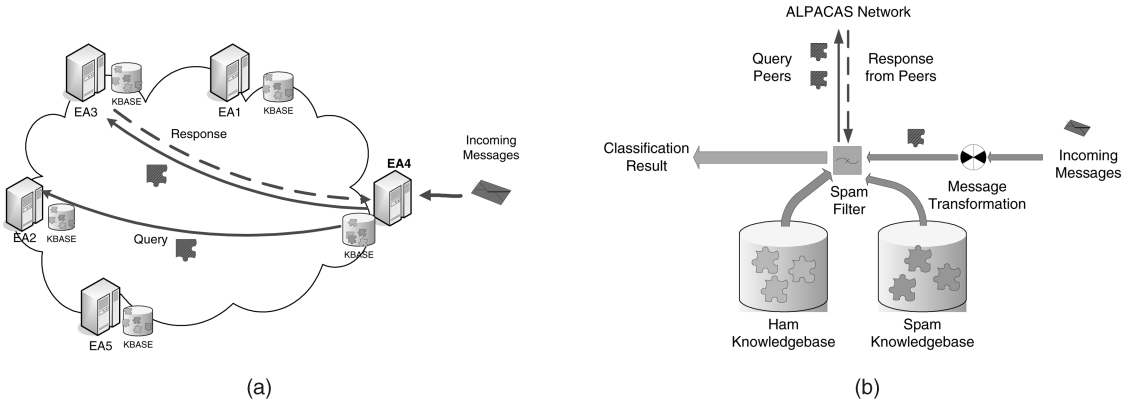


Fig. 1. ALPACAS system overview. (a) ALPACAS network. (b) Internal mechanism of EA_4.

These two drawbacks, namely vulnerability toward camouflage attacks and potential risk of privacy breaches, highlight the need for better collaborative mechanisms that are not only resilient toward minor differences among messages but are also robust against inference-based privacy compromises.

Anti-spam collaboration has also been proposed in the form of spam detection using e-mail social network [21]. These approaches are orthogonal to the work presented in this paper and can be used to further enhance the effectiveness of our system.

2.3 Privacy-Aware Data Management

Recently, there has been considerable research on privacy and trust issues in data management [22], [23], [24], [25], [26]. Data perturbation [22] and data anonymization [27], [23], [24] are the two basic approaches for ensuring privacy of relational data. Researchers have also proposed various privacy-aware schemes for sharing information among independent databases [28]. Further, the problems of privacy-preserving query computation and data mining have also received considerable research attention [29], [30]. However, most of these schemes cannot be used for collaborative spam filtering application as the underlying data is essentially textual in nature.

3 The ALPACAS Anti-Spam Framework

We present the ALPACAS framework to address the design challenges of the collaborative anti-spam system.

- **Challenge 1.** To protect e-mail privacy, it is obvious that the messages have to be encrypted. However, in order for the collaboration to be effective, the encryption mechanism has to satisfy two competing requirements: 1) The encryption mechanism has to hide the actual contents for privacy protection and 2) it should retain important features of the message so that effective similarity comparison can still be performed on the encrypted messages.
- **Challenge 2.** To avoid inference-based privacy breaches, it is necessary to minimize the information revealed during the collaboration process. However, the lesser the information conveyed, the harder it is to perform meaningful similarity comparisons.

Accordingly, the ALPACAS framework includes two unique components, namely *feature-preserving fingerprint* and *privacy-preserving protocol* to address the above challenges, respectively. In addition, in the interests of scalability, we design a *DHT-based architecture* for distributing ham/spam information among the collaborating entities.

The ALPACAS framework essentially consists of a set of collaborative anti-spam agents. An e-mail agent can either be an entity that participates in the ALPACAS framework on behalf of an individual user, or it may represent an e-mail server having multiple users. Without loss of generality, in this paper, we assume that the e-mail agents represent individual users. Each e-mail agent of the ALPACAS framework maintains a spam knowledge base and a ham knowledge base, containing information about the known spam and ham e-mails. Fig. 1a shows the e-mail agent EA_4 querying two other collaborative agents with *partial* information of an incoming message for the purpose of classification. Fig. 1b illustrates the internal mechanism of each e-mail agent: Upon receiving an e-mail, the respective e-mail agent transforms the message into a feature digest. It then uses *part* of the feature digest to query a few other e-mail agents to check whether they have any information that could be used for classifying the e-mail. Based on the responses from these agents and its local knowledge base, a simple method to classify an e-mail is presented in Section 3.2.

3.1 Feature-Preserving Fingerprint

In our approach, the fingerprint of an e-mail is a set of digests that characterize the message content. The set of digests is referred to as the *transformed feature set* (TFSet) of the e-mail. The individual digests are called the *feature elements* (FEs). The TFSet of a message M_a is represented as $\mathrm{TFSet}(M_a)$. In the following sections, we will discuss how to generate TFSet and how to further enforce the privacy preservation.

3.1.1 Shingle-Based Message Transformation

Our feature-preserving fingerprint technique is based upon the concept of Shingles [31], which has been used in a wide variety of Web and Internet data management problems, such as redundancy elimination in Web caches and search

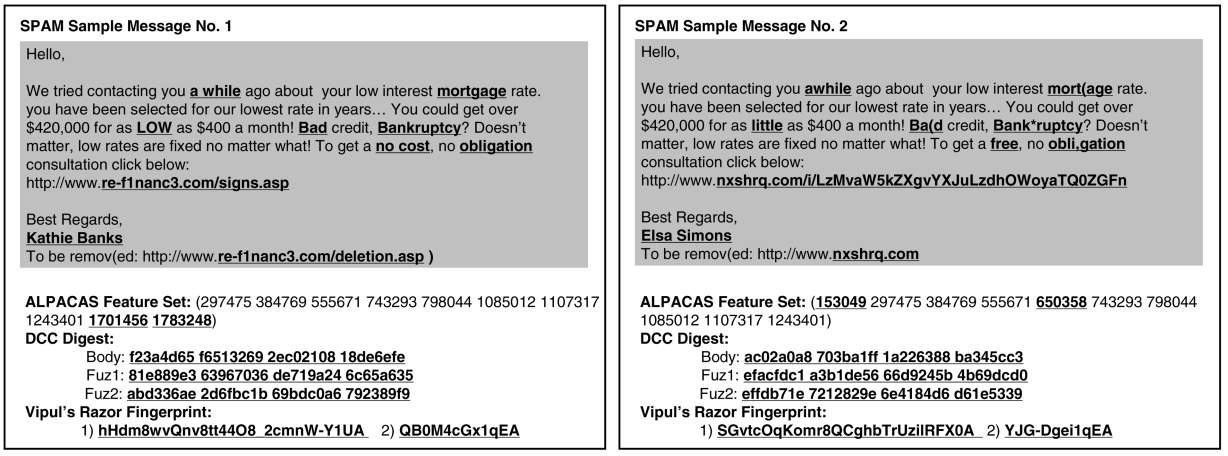


Fig. 2. ALPACAS feature sets, DCC, and Razor digests for two spam e-mails (texts in bold font indicate differences).

engines, and template and fragment detection in Web pages [32], [33].

Shingles are essentially a set of numbers that act as a fingerprint of a document. Shingles have the unique property that *if two documents vary by a small amount, their shingle sets also differ by a small amount*.

Fig. 2 presents an example to illustrate the strength of this feature-preserving fingerprint technique. The figure shows two real spam e-mails that are very similar to each other. The spammers have deliberately mutated one of the e-mails through word and letter substitutions to obtain the other. The figure shows the TFsets of the two e-mails. For comparison purposes, we also indicate the results of the MD-5, Vipul's Razor, and the DCC transformations on the two e-mails. For MD-5, Vipul's Razor, and DCC, the hash digests of the two e-mails are totally different from each other, whereas the shingle sets of the two e-mails retain a high degree of similarity that 80 percent of the TFsets of both spam e-mails are the same.

To generate a TFset of a message M, we use a sliding window algorithm, in which a window of some predetermined length (W) slides through the message. At each step, the algorithm computes a Rabin fingerprint [34] of W consecutive tokens (a token could be either a single word or character, and we use character-based token throughout this paper) that fall within the window. Each fingerprint is in the range $(0, 2^K - 1)$, where K is a configurable parameter. For a message with X tokens, we obtain a set of $X - W + 1$ fingerprints. Of these, the smallest S fingerprints are retained as the (W, S) TFset of M, because using a subset of the fingerprints that represent partial information of M provides more privacy protection than using the entire set of fingerprints. We represent (W, S) TFset of a message M as $\mathrm{TFSet}_{(W,S)}(M)$. The similarity between two messages M_a and M_b can be calculated as $\frac{|\mathrm{TFSet}_{(W,S)}(M_a) \cap \mathrm{TFSet}_{(W,S)}(M_b)|}{|\mathrm{TFSet}_{(W,S)}(M_a) \cup \mathrm{TFSet}_{(W,S)}(M_b)|}$.

In consideration of the privacy preservation, the message transformation uses a Rabin fingerprint algorithm, which is a one-way hash function such that it is computationally infeasible to generate the original e-mail from its TFset However, it is possible to infer a word or a group of words from an individual feature value. The privacy protection requires multiple levels of defenses. In Section 3.1.2, we present our privacy enhancement.

The transformation from message to feature set can be based on words rather than characters, i.e., the sliding window is over the W consecutive words rather than bytes. We choose to use character-based token selection because it is more general than word-based token selection and can be easily implemented. It considers important features, such as message layout symbols, rather than just the text. The character-based selection is also better suited for short messages, such as Picospam and comment spam.

3.1.2 Term-Level Privacy Preservation

Term-level privacy breach is defined as an FE uniquely identifying a word or a group of words, and an e-mail agent could infer a phrase or a sentence out from a feature with a reasonable probability if the agent had come across a previous message whose TFset contained the same feature value. For example, a term "$99,999" corresponds to a shingle value of 16,067,109. If a recipient of message M_a knows that the encryption of message M_b contains a common shingle value of 16,067,109, he can immediately infer that M_b also contains the term "$99,999."

One approach to mitigate the possibility of inferring a word or a group of words is to shuffle the tokens (i.e., bytes) of the original e-mail and compute TFset on the shuffled e-mail. Though this is expected to accomplish term-level privacy compromise, arbitrary and large-scale shuffling can destroy the e-mail features, thereby affecting the spam filtering accuracy.

To shuffle the e-mail content in an acceptable manner, our feature-preserving fingerprint scheme adopts a *controlled shuffling* strategy wherein the tokens are shuffled in a predetermined format. Further, the position of a token after shuffling is always within a fixed range of its original position.

Specifically, the controlled shuffling scheme works as follows: The e-mail text is divided into consecutive *chunks* of tokens. Each chunk consists of z consecutive tokens of the e-mail text, where z is a configurable parameter. The tokens in each chunk are shuffled in a predetermined manner, whereas

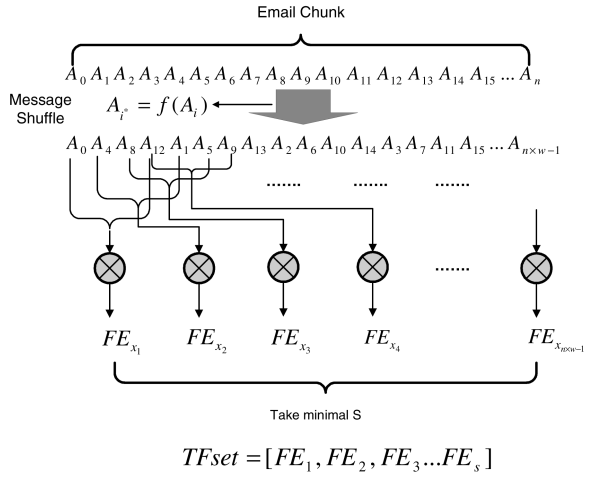


Fig. 3. Feature-preserving fingerprint technique.

the ordering of the chunks within the e-mail text remains unaltered. Concretely, each chunk is further divided into y *subchunks* (we assume that y is a factor of z). The tokens within an arbitrary chunk CK_h are shuffled such that the token at $r\text{th}$ position in the $s\text{th}$ subchunk (this is the token at the index $(s \times \frac{z}{y}) + r$) in the chunk CK_h) is moved to $(r \times y + s)\text{th}$ position within CK_h.

Fig. 3 illustrates the feature-preserving fingerprint technique (controlled replication of the e-mail text followed by shingle computation) on an e-mail chunk, with the subchunk size being set to 4. We demonstrate the effectiveness of this controlled shuffling strategy in Section 4.7.

Suppose two different messages contain an identical term, by shuffling the term, the rendered text could be different. Thus, it could make the FE generated from the shuffled text different. We expect this controlled shuffling scheme to reduce the term-level privacy breach. A comprehensive study on this subject will be done in our future work.

3.1.3 Illustration on Real Examples

Let us revisit two real spam examples in Fig. 2 to illustrate the strength of our feature-preserving transformation. These two spam e-mails are delivered to different recipients at different time. To avoid being caught by digest-based approach, a spammer deliberately modifies the content by replacing word, character, or http links. Suppose the window size is 8, by using our character-based transformation, message 1 will generate 440 FEs while message 2 will have 444 FEs disregarding the duplicated FEs. The number of different FEs between these two messages is around 100. Thus, the majority of the FEs is common because of the sliding window feature of shingle-based technique. In the example, the eventual TFset size is 10, so that only 10 smallest FEs out of 440 FEs are picked to represent message 1, and 10 smallest FEs out of 444 FEs are picked to represent message 2. The resulted TFSets demonstrate 80 percent in common indicating the significant similarity.

3.1.4 Possible Attacks

In order to make two spam e-mails dissimilar, spammer has to make the generated TFsets different. There are two theoretically possible ways: 1) It takes moderate efforts for spammers to attach texts corresponding to small FE values to the messages. By doing this, spammers try to make the messages dissimilar as if they have never been seen by our system, because our system transforms the text into FEs and measures the similarity with only the smallest S FEs. However, this attack results in the messages not similar to any spam or ham. To further classify the messages from this attack, one possible solution is to design an algorithm that compare the full range of FEs so that the similarity comparison would not be biased to the use of small FE values. 2) To attack similarity comparison using full-range FEs, a spammer would make changes everywhere in the message to make FEs totally different but still keep the original meaning. However, this is relatively hard to achieve. The more the message is modified, the harder the spam e-mail can maintain the original content.

3.2 Privacy-Preserving Collaboration Protocol

Feature-preserving fingerprint is just one level of privacy protection, the amount of information exchanged during collaboration can be further controlled for stronger privacy protection. In particular, we design the collaborative anti-spam system equipped with privacy-aware message exchange protocol based on the following spam/ham dichotomy that *revealing the contents of a spam e-mail does not affect the privacy or confidentiality of the participants, whereas revealing information about a ham e-mail constitutes a privacy breach.*

Our protocol works as follows: When an agent EA_j receives a message M_a, EA_j computes its $\text{TFSet}: \text{TFSet}(M_a)$. It then sends a query message to other e-mail agents in the system to check whether they can provide any information related to M_a. However, instead of sending the entire $\text{TFSet}(M_a)$ as the query message to all agents, EA_j sends a small subset of $\text{TFSet}(M_a)$ to a few other e-mail agents (the e-mail agents to which the query is sent is determined on the basis of the underlying structure; see Section 3.3). The subsets of $\text{TFSet}(M_a)$ included in the queries sent to various other e-mail agents need not be the same (our architecture optimizes the communication costs by sending nonoverlapping subsets to carefully chosen e-mail agents).

An e-mail agent that receives the query, say EA_k, checks its spam and ham knowledge bases looking for entries that include the feature subset that it has received. A feature set is said to *match* a query message if the set contains all the FEs included in the query. Observe that there could be any number of entries in both spam and ham knowledge bases matching the partial feature set. For each matching entry in the *spam knowledge base*, EA_k includes the *complete* TFSet of the entry in its response to EA_j. However, for any matching *ham entries*, EA_k sends back a small, *randomly selected part* of the TFSet. Fig. 4 illustrates our privacy-preserving collaboration protocol. In this figure, the agent EA_4 sends a query with the FE 815033 to EA_7, which responds with a complete feature set of a matching spam e-mail and a partial feature set of a matching ham e-mail.

At the end of the collaboration protocol, EA_j would have received information about any matching ham and spam e-mails (containing the feature set of the query) that have been received by other members in the collaborative

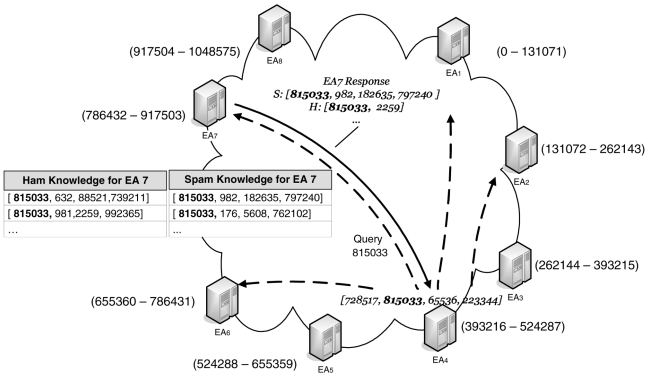


Fig. 4. ALPACAS protocol: query and response.

group. For each matching spam e-mail, EA_j receives its complete TFSet. For each matching ham e-mail, EA_j receives a subset of its TFSet. EA_j now computes the ratio of $MaxSpamOvlp(M_a)$ to $MaxHamOvlp(M_a)$ and decides whether the M_a is spam or ham. $MaxSpamOvlp$ is the maximum overlaps between the TFSet of the query message and the TFSets of all the matching spam e-mails, and $MaxHamOvlp$ is similarly defined. In this paper, we use a simple classification strategy that is described as follows:

$$Score = \frac{1 + MaxSpamOvlp(M_a) - MaxHamOvlp(M_a)}{2}. \quad (1)$$

If the score is greater than a configurable threshold λ, M_a is classified as spam. Otherwise, it is classified as ham. We provide the pseudocode for the system operation in Fig. 5.

3.3 Distributed System Structure

As in many other distributed systems, the underlying architecture has a strong inference on the efficiency, scalability, and performance of the ALPACAS system. However, an aspect that is unique to this problem is that the underlying architecture can also have a significant impact on the privacy of the participating e-mail agents. For example, sending queries to too many e-mail agents increases the risk of inference-based privacy breaches.

A naive approach for designing the ALPACAS system is to use a flat and unstructured organization [35] in which every agent maintains its knowledge base of the spam information that it has received. In this case, an agent would need to query all other collaborative agents to classify each incoming message. Thus, the flat structure is inefficient and unscalable. It also has high possibility of privacy breaches as the FEs of a message are virtually delivered to every participant in the system.

We design an efficient and scalable structure for the ALPACAS prototype, which also minimizes the chances of inference-based privacy breaches. Our prototype structure is based upon the following design principle: *A query should be sent to an e-mail agent only if it has a reasonable chance of containing information about the e-mail that is being verified. Contacting any other e-mail agent not only introduces inefficiencies but also leads to unnecessary exposure of data.*

The proposed prototype structure is based on the distributed hash table (DHT) paradigm [36], [37]. In this

INPUT:
EMAIL EM_a, Threshold λ

OUTPUT:
Ham or Spam

PROCEDURE:
Transform the incoming EM_a into a feature set *TFset(EM_a)*.
Initialize *queryResultSet* to null
for each element in *TFset(EM_a)*
    rendezvousResult=queryRendezvous(element)
    insert *rendezousResult* into *queryResultSet*

Initialize *MaxHamOvlp(EM_a)* and *MaxSpamOvlp(EM_a)* to 0
for each *TFset(EM_{ham})* in *queryResultSet*
    measure *HamOvlp(EM_a, EM_{ham})*
    if *HamOvlp(EM_a, EM_{ham}) > MaxHamOvlp(EM_a)*
        MaxHamOvlp(EM_a) = HamOvlp(EM_a,EM_{ham})

for each *TFset(EM_{spam})* in *queryResultSet*
    measure *SpamOvlp(EM_a, EM_{spam})*
    if *SpamOvlp(EM_a, EM_{spam}) > MaxSpamOvlp(EM_a)*
        MaxSpamOvlp(EM_a) = SpamOvlp(EM_a,EM_{spam})

Email score:
$EM_{score} = \frac{1+MaxSpamOvlp(EM_a)-MaxHamOvlp(EM_a)}{2}$
if $EM_{score} > \lambda$
    return *Spam*
else
    return *Ham*

Fig. 5. Pseudocode for e-mail classification operation.

DHT-based structure, each e-mail agent is allocated a range of FE values. An e-mail agent EA_j is responsible for maintaining information about all the e-mails (received by any e-mail agent in the system) whose TFSet has at least one FE in the range allocated to it. Specifically, if there are N e-mail agents in the collaborative group, the range $(0, 2^K - 1)$ (recall that all FEs lie within this range) is divided into N nonoverlapping consecutive regions represented as $\{(MinF_0, MaxF_0), (MinF_1, MaxF_1), \ldots, (MinF_{N-1}, 2^K - 1)\}$, where $(MinF_j, MaxF_j)$ denotes the subrange allocated to the e-mail agent EA_j. EA_j maintains information about every spam and ham e-mail that has at least one FE between $MinF_j$ and $MaxF_j$ (inclusive of both endpoints). For each such spam e-mail, EA_j stores the entire TFSet in its spam knowledge base. For ham e-mails, EA_j stores a subset of the e-mail's TFSet. If the FE value Ft falls within the subrange allocated to EA_j (i.e., $MinF_j \leq Ft \leq MaxF_j$), then EA_j is called the *rendezvous agent* of Ft. The set of rendezvous agents of all the FEs of M_a is called M_a's rendezvous agent set. The spam and ham knowledge bases at a rendezvous agent are indexed by the FE that falls within the agent's subrange. Fig. 4 illustrates an ALPACAS prototype with eight agents and FEs in the range of (0, 1,048,575).

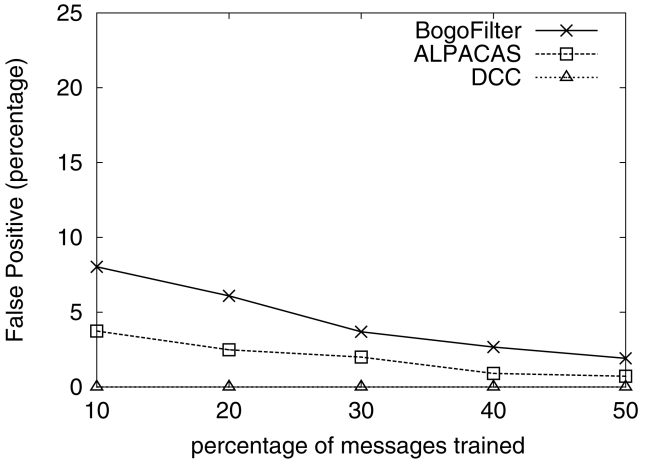


Fig. 6. False positive percentages of ALPACAS, Bogofilter, and DCC.

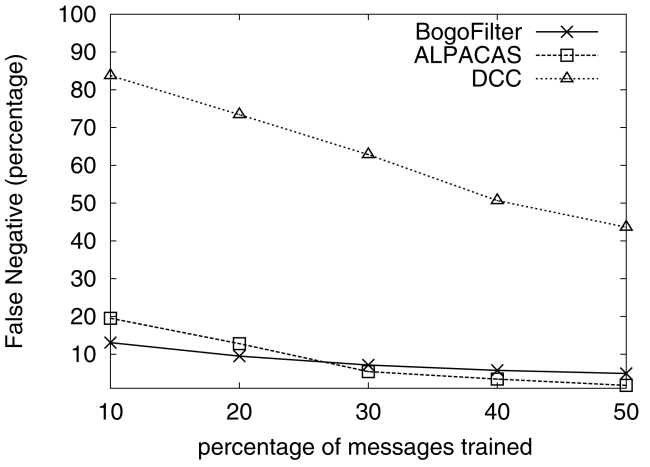


Fig. 7. False negative percentages of ALPACAS, Bogofilter, and DCC.

The presented DHT structure is only for proof of concept. This paper focuses on the feasibility of collaboration with transformed messages and we expect that a more sophisticated and robust P2P structure is applied in a real deployment.

4 Experiments and Results

In this section, we compare ALPACAS with two popular spam filtering approaches, namely Bayesian filtering and simple hash-based collaborative filtering. We use Bogofilter 1.1.1 [4] and DCC 1.2.74 as the representatives of these two approaches, respectively. As most other Bayesian filters, Bogofilter calculates a score (spamminess) for each message. The message is classified as a spam if its spamminess is greater than or equal to a preset threshold (μ) and vice versa. On the other hand, the DCC bases its decision on the number of times the e-mail corresponding to a particular hash value have been reported as spam. If this *spam count* of the hash value corresponding to incoming e-mail exceeds a threshold, the e-mail is classified as spam; otherwise, it is classified as ham.

We conduct a comprehensive study on the accuracy comparison between ALPACAS and Bogofilter for the entire range of the threshold. For other performance measurements, the default threshold for both is set to 0.5. Since DCC is strongly biased to a low false positive rate, we set the DCC threshold to 1, which gives the best false negative rate as shown in Fig. 7.

4.1 Experimental Setup

The data sets used in our experiments are derived from two publicly available e-mail corpus, namely TREC 2005 e-mail corpus [14] and the SpamAssassin e-mail corpus [10]. To simulate the collaboration among recipients, we categorize the e-mails in the TREC corpus, which are the real e-mails from Enron according to their target addresses ("To:" and "cc:" fields) to obtain 67 e-mail sets, each corresponding to the e-mails received by one individual. Half of each e-mail set including ham and spam are used for training, and the remainder is used for testing. In the experiment, we also assume that each individual can have a preclassified e-mail corpus (spamAssassin corpus) as the initial knowledge base. Each individual incrementally feeds the knowledge base with a fraction of his e-mail set (TREC) categorized for the training purpose. We measure the accuracy based on all the incoming messages to the system composed of a number of participants. We compare the results among different systems including Bogofilter, DCC, and ALPACAS adopted by the participants. The difference is that Bogofilter makes classification decision based on each individual's local message information, while DCC and ALPACAS rely on the shared message information among all the participants.

4.2 Performance Metrics

We use the standard metrics to measure the spam filtering accuracy. A ham e-mail that is classified as spam by the filtering scheme is termed as a *false positive*. The false positive percentage is defined as the ratio of the number of false positive e-mails to the total number of actual ham e-mails in the data set used during the testing phase. The false negative percentage is analogously defined.

Currently, there is no available metrics to measure the privacy of collaborative anti-spam systems. In this paper, we first define the *message-level privacy breach percentage* as follows: A ham e-mail M_a is said to have suffered a privacy compromise if an e-mail agent that is not a recipient of M_a discovers its contents. *Message-level privacy breach percentage is defined as the ratio number of test ham messages suffering privacy compromises to the total number of test ham messages.*

The communication overhead of the system is quantified through the *per-test communication cost* metric, which is defined as the total number of messages circulated in the system during the entire experiment.

4.3 SPAM Filtering Effectiveness

In the first set of experiments, we study the effectiveness of ALPACAS approach in filtering traditional spam messages (as captured by the testing data sets). Fig. 6 shows the false positive percentages of the Bogofilter, ALPACAS, and DCC schemes when the size of the training set employed by each agent increases from 10 percent to 50 percent of the total messages in its e-mail set. Fig. 7 indicates the false negative rates for the same experiment.

In general, as we expect, ALPACAS has strong feature-preserving capabilities and demonstrates a better accuracy than Bogofilter when there are enough e-mail resources

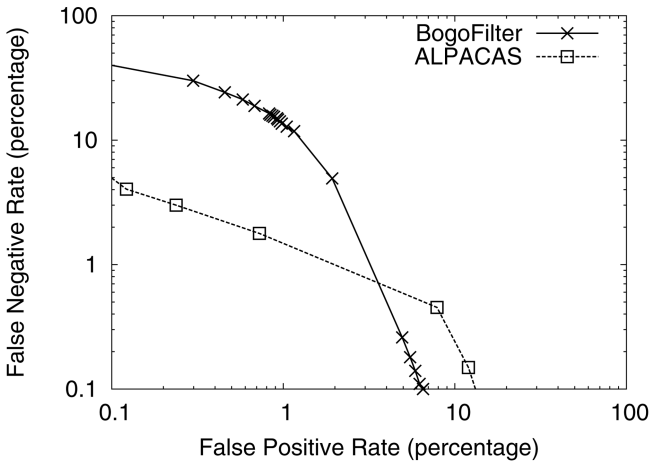


Fig. 8. System overall accuracy (DCC is not displayed because its FP is 0).

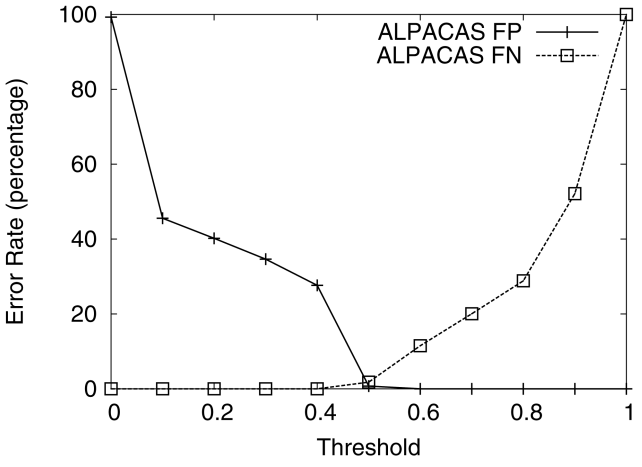


Fig. 9. Effects of threshold parameter on ALPACAS.

shared in the network. Fig. 6 shows that ALPACAS always performs a better false positive percentage than the Bogofilter. For the false negative percentage shown in Fig. 7, ALPACAS is better than Bogofilter after around 27 percent of the messages in the e-mail sets are employed during the training phase. And, ALPACAS shows about 60 percent lower false negative percentage than that of the Bogofilter when 50 percent of the messages in the e-mail sets are used for training.

The results also indicates that the essence of the collaboration is knowledge sharing. When the size of the training sets employed at the individual agents is small, ALPACAS does not demonstrate a better false negative rate than the Bogofilter. It is also natural that transformed message is less effective than the original message. Furthermore, DCC performs much worse for the false negative percentage than the other two schemes. Note that the false negative percentages of DCC are an order of magnitude higher than our approach.

All the ALPACAS, DCC, and Bayesian schemes are threshold-based approaches, so finding the appropriate threshold to achieve both low false positive and false negative rates is the key to the success of these approaches. We obtain results from previous experiment when 50 percent of the e-mails in its e-mail set are used during the training phase. We vary the threshold parameters of the two schemes and collect the false positive and false negative percentages. In Fig. 8, we plot the results of the experiment with false positive percentages on the x-axis and the false negatives on the y-axis.

The results show that neither of the approaches outperforms the other at all false positive percentage values. However, the ALPACAS approach yields significantly better false negative results than the Bogofilter for the normally preferred false positive range. Generally, users have a much lower tolerance of false positives than false negatives, and anything more than 1 percent false positives is usually considered unacceptable.

Fig. 9 shows the effects of threshold parameter (λ) on the false positive and the false negative percentages of the ALPACAS approach. As we expect, the false positive percentages decrease with increasing values of λ, whereas the false negative percentages show a corresponding increase. From this figure, we conclude that the ALPACAS approach yields best performance when λ is around 0.5.

In summary, ALPACAS has an overall comparable accuracy to the current approaches such as Bogofilter. It has advantages over Bogofilter when low false positive is preferred. Notice that, even with the same accuracy results, a collaborative filter is often preferred because of its resistance to camouflage attacks, which is presented in Section 4.4.

4.4 Robustness against Attacks

In this section, we evaluate the robustness of the ALPACAS approach against two common kinds of camouflage attacks, one is *good word attack* and the other is *character replacement attack*. We compare the results with those of Bayesian and DCC approaches.

4.4.1 Behavior of ALPACAS under Normal Attacks

In the first experiment of this series, we emulate the good word attack by appending words that generally appear in ham messages in the test set. The good words are selected randomly from a good word database created from the labeled ham data. We vary the amount of appended words in the range of 0 percent to 100 percent of the original e-mails' word count and we call it *degree of attack*. The experimental setup consists of 67 agents with each agent employing 50 percent of the messages in its e-mail set during the training phase.

Fig. 10 shows the false negative rate of Bogofilter, DCC, and the ALPACAS approach at various degrees of attack. False positive results are not presented because they are not affected by the attacks. The false negative percentages of the ALPACAS and Bogofilter are very low when the degree of attack is less than 5 percent. However, the performance of the Bogofilter degrades drastically as the degree of attack increases, whereas the false positive percentage of the ALPACAS approach increases by very small amounts. For example, when the amount of good words introduced is around 80 percent, the false negative rate of Bogofilter is close to 100 percent, whereas it is around 7 percent for the ALPACAS scheme. The performance of DCC is very poor for all its different forms of checksums even at very low degrees of attack. This is because of the nature of its hashing

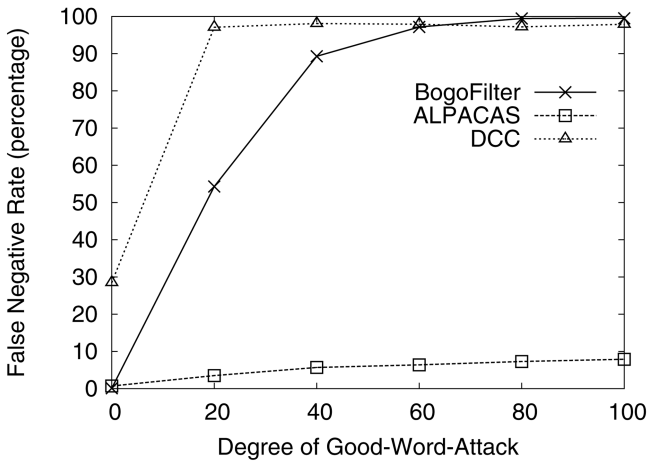


Fig. 10. System robustness against good word attacks.

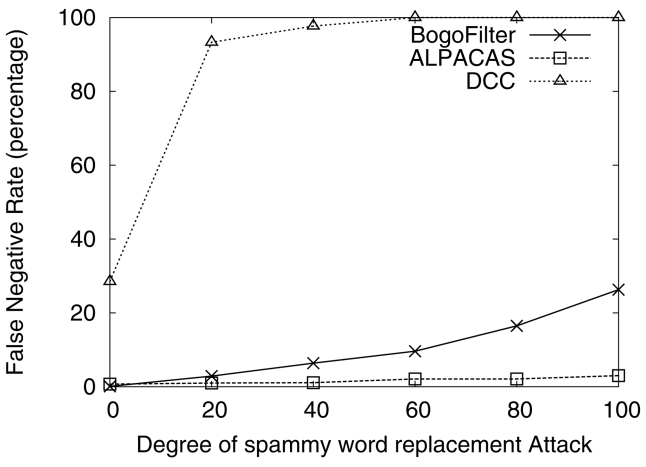


Fig. 11. System robustness against character replacement attacks.

mechanism, which maps similar (but not identical) messages into two totally different hash values.

In the second experiment of this series, we study the resilience of the ALPACAS, Bogofilter, and DCC schemes toward another common type of attack, which we call *character replacement attack*. In this attack, the spammer replaces a few characters of certain fraction of words that are highly likely to be present in spam e-mails (henceforth, we refer to these words as "spammy words"). The spammer attempts to reduce the spam weight (weight indicating the probability that the e-mail is a spam) assigned by filters to the e-mail. E-mails containing "Vi@gra" instead of "Viagra" are examples of character replacement attacks. In order to emulate this attack, we first create a spam dictionary. For each e-mail in the corpus, we extract the words that appear in the spam dictionary. We then replace a few characters of a certain randomly selected fraction of the words in the spam list. The ratio of the number of changed words to the total number of words in the e-mail that appear in the spam dictionary is called the *degree of attack*.

We then measure the filtering effectiveness of the three anti-spam schemes. The setting is similar to that of the previous experiment. Fig. 11 shows the false negative percentage of the three schemes when the percentage of spam words that were modified in each e-mail varies from 0 percent to 100 percent. As the degree of attack increases, the effectiveness of Bogofilter deteriorates. When 100 percent of spammy words are modified, the false negative percentage is as high as 27 percent. In contrast, the false negative percentage of the ALPACAS system is 3 percent even when 100 percent of spammy words are modified. The DCC again performs very poorly even at low degrees of attack.

4.4.2 Performance of ALPACAS under Severe Attacks

We investigate the limitation of ALPACAS under "good word" attacks by measuring the filtering accuracy (false negative) with a high degree of good word attacks. The result is presented in Fig. 12, which shows that with four times the number of good words, ALPACAS's false negative rate degrades to 27.1 percent, and with about 30 times of good words, the ALPACAS system would not detect any of the camouflaged spam messages.

Although these high-degree "good word" attacks are possible, the attacks are less likely to be effective because of two factors. First, if these good words are directly added to the spam text, they would significantly affect the readability of the spam and reduce the effectiveness of the spam. More likely, spammers will try to hide these good word attachments with tricks such as text in background colors or HTML comments. The actions that hide a major part of the message from a recipient are likely to become the features to identify spam and have been seen as effective spam detection features used by current filters [16].

It is also possible for the spammer to insert strings that cause fingerprints to collide with the fingerprints of some ham messages. The goal is to increase the spam message's similarity to some ham message. In fact, spammers have already applied this to attack statistical learning filters that inspect the original text. A privacy-aware filtering system like ALPACAS does not eliminate this type of attack. However, adding a small number of features that collide with ham messages would not eliminate the message's common features that are shared with some spam message. Spammers would have to add a large amount of unique ham-like features to reduce the similarity to previous spam.

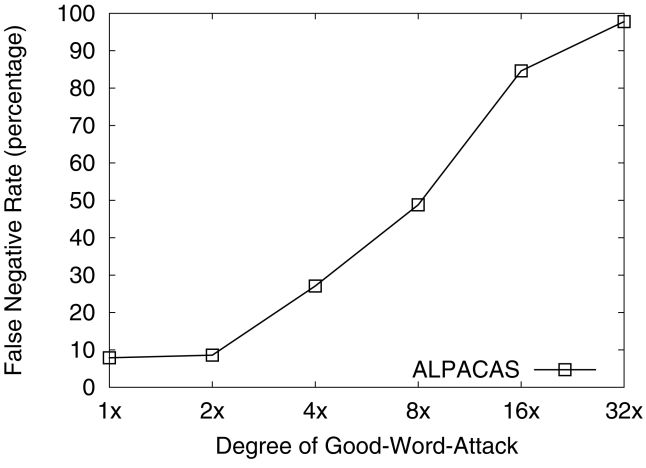


Fig. 12. The limitation of ALPACAS against good word attacks.

TABLE 1
Privacy Breach (**Metric 1**): Effectiveness of Partial Sets for Various Number of Agents

Privacy Breach		Number of Feature Elements used in a Query						
		1	2	10	20	30	40	50
AgentNum	100	0.17%	0.25%	0.28%	0.30%	0.31%	0.32%	0.33%
	200	0.09%	0.13%	0.16%	0.17%	0.18%	0.19%	0.20%
	300	0.06%	0.07%	0.08%	0.09%	0.09%	0.10%	0.11%
	400	0.05%	0.06%	0.07%	0.07%	0.07%	0.07%	0.07%
	500	0.06%	0.07%	0.08%	0.08%	0.09%	0.09%	0.09%
	600	0.04%	0.05%	0.06%	0.06%	0.07%	0.07%	0.08%

It is essentially similar to those attacks with high degree of good word attacks except it requires additional efforts from spammers to search fingerprint collisions.

4.5 Privacy Awareness of the ALPACAS Approach

One major design consideration of the ALPACAS approach is preserving the privacy of the e-mails and their recipients. To measure the privacy breaches, we emulate the following model for privacy compromises. When a rendezvous agent EA_i obtains a part of the TFSet of an e-mail M_a (either for querying or for publishing), EA_i collects all the ham e-mails received by it that match the part of the feature set that has been sent to it. In the absence of any further information, EA_i selects one of these matching ham e-mails, say M_b as its guess. In other words, EA_i guesses the contents of the e-mail M_a to be similar to that of M_b. If the guess is correct (the contents of M_a are indeed similar to those of M_b), then we conclude that a privacy breach has occurred. We count such privacy breaches to calculate the message-level privacy breach percentage.

The privacy breach also relates to how much information is conveyed during the collaboration. We consider three different query policies in our experiment: 1) query with minimal feature set, 2) query with full feature set, and 3) query with partial feature set. To further reduce the content breach possibility, we only share spam knowledge across the collaborative network.

In this work, we introduce two metrics to measure the privacy:

- **Metric 1.** When a rendezvous agent EA_i obtains a part of the TFSet of an e-mail M_a, EA_i takes a guess no matter whether it has matched feature sets in its ham e-mails. We regard the number of total guesses in the ALPACAS network as G. If EA_i has m matched feature sets in its ham e-mails, and EA_i has an exact ham message as M_a, the probability p_i to guess that it has the same message as M_a is $\frac{1}{m}$; otherwise, p_i is 0. Suppose the number of the agents in the ALPACAS is N, the overall privacy breach rate is measured as $\frac{\Sigma_{i=1}^{N} p_i}{G}$.
- **Metric 2.** The privacy is measured in a similar way with the only difference that EA_i only takes guesses whenever there is a match for the feature set of M_a being queried. So, the total necessary guesses in the ALPACAS network is G', $G' \leq G$. And, the overall privacy breach rate is measured as $\frac{\Sigma_{i=1}^{N} p_i}{G'}$. In the rest of this section, we present the experimental results for these two metrics.

4.5.1 Privacy Awareness Using Metric 1

Table 1 shows the message-level privacy breach percentages of the ALPACAS approach as the number of collaborating agents varies from 100 to 600 for the three query policies. Since the TREC data set only contains e-mails received by 67 individuals, we split the e-mail set corresponding to each user into 10 equisized trace files. Each of these trace files drives an e-mail agent. The number of FEs in the TFSet of each e-mail is 50, and 50 percent of the e-mails in each trace is used during the training phase.

The results show that the privacy breaches are very rare for all three modes of the ALPACAS approach. For metric 1, the privacy breach rate depends on the chances that a querying message in the form of a partial feature set can find an identical feature set in the peer neighbors. Normally, the privacy breach percentages go down as the number of agents in the system increases. This can be explained as follows: When the number of e-mail agents in the system increases, the range of DHT values allocated to each e-mail agent decreases. Thus, a rendezvous agent is unlikely to have received a similar e-mail in the recent past. There can be exceptions, for example, in Table 1, when the node number is 400, it shows a lower privacy breach rate. It is because when we split the data sets into 10 equal-sized trace files and assemble them into 400 separate agents, it happens to have some identical files in the same agent, in which case it reduces the possibility of privacy breach.

Although with an overall low privacy breach, the reduction of privacy breach by using partial sets is not as significant as we expected. We ascribe this behavior to the small number of e-mail instances in our testing set when compared to the large feature set space. To demonstrate the reduction of privacy breach by using partial sets, we present results using our second metric in Section 4.5.2.

4.5.2 Privacy Awareness Using Metric 2

Table 2 demonstrates privacy results using our second metric. The result shows that when the full feature set is used as the query, 100 percent privacy breach is introduced wherever there is a privacy breach threat. However, by using only one shingle value in the feature set, the privacy breach is well controlled below 1 percent. The privacy

TABLE 2
Privacy Breach (**Metric 2**): Effectiveness of Partial Sets for Various Number of Agents

Privacy Breach		Number of Feature Elements used in a Query						
		1	2	10	20	30	40	50
AgentNum	100	0.89%	12.3%	42.5%	62.7%	75.5%	83.4%	100%
	200	0.80%	11.8%	38.1%	58.3%	73.6%	83.5%	100%
	300	0.64%	8.6%	33.8%	53.1%	70.4%	78.6%	100%
	400	0.56%	9.7%	39.2%	58.1%	77.8%	86.3%	100%
	500	0.95%	15.3%	51.8%	72.8%	86.1%	90.4%	100%
	600	0.62%	8.4%	29.0%	47.5%	68.5%	78.3%	100%

breach rate increases as the amount of the feature set exposed in the query increases. This is because the more information exposed, the greater the chance the message is guessed correctly. We also find that when two shingle values are used, the privacy breach rate increases sharply. We ascribe it to the small number of e-mail instances compared to a large feature set space.

In general, for our second metric, the impacts to the privacy breach rate come from the number of identical messages delivered to multiple recipients. To further control the privacy breach, we plan to continue our study by two means: one is to experiment with various sizes of data sets and feature set spaces; the other is to use feature range in the query rather than the exact feature value, with the hope to further hide the real feature value for the purpose of privacy protection.

4.5.3 The Overall Privacy and Accuracy Trade-Off

Our study of the relationship between feature set size and privacy protection as well as filter accuracy illustrates a fundamental conflict of privacy-aware spam filtering, which is the privacy and accuracy trade-off. Intuitively, a spam filter's accuracy should increase as more feature information (i.e., more FEs) is being used; whereas in the query and response, more feature information would result in a higher risk of privacy breaches. Our study, however, shows that the privacy and accuracy trade-off is more subtle than this intuition, and the exact trade-off behavior depends on the detail of the privacy control mechanism.

In ALPACAS, the risk of privacy breach is controlled by two levels of mechanism: the number of FEs used in the query and the size of feature set that represents a message. For a given feature set size, a range of privacy protection can be achieved by using a partial set in queries. Smaller partial sets would contain fewer FEs in each query and thus provide better privacy protection to the message. Since the responses suppose to contain information about all spams that match the queries, the privacy protection mechanism (by sending partial sets) does not affect the filter accuracy. This is because all the spam messages that are similar to the complete feature set would have been already included in responses to queries using partial sets. Therefore, the intuition is not true for the mechanism of privacy protection by using partial sets in queries. However, in the situation of privacy protection by controlling the feature set size directly, a small feature set often means good privacy protection but would result to a lower filtering accuracy. The experimental results, as shown in Figs. 15 and 16, have confirmed this by illustrating accuracy versus the feature set size.

As indicated in Fig. 14, filter accuracy has an inverse relationship with the possibility of privacy breach (measured in terms of Metric 2). Obviously, no collaboration is possible if the participating agents require absolute privacy protection $(\text{privacy breach possibility} = 0)$. This situation is equivalent to stand-alone filtering, and thus, the accuracy is rather poor. However, the accuracy improves drastically for a small trade-off in privacy breach possibility. For example, the false negative rate drops to less than 15 percent for privacy breach possibility of 10 percent. However, beyond a certain point $(\text{false negative rate} = 5 \text{ percent})$, even a multi-fold increase in privacy breach possibility only provides a marginal improvement in filter accuracy. The exact trade-offs and the optimal parameter settings certainly depend on the details of system configuration and privacy protection algorithms. Nevertheless, this result indicate that without introducing high risks of privacy breaches, the ALPACAS system achieves filtering accuracies that are close to its best.

4.6 Communication Overheads of the ALPACAS Approach

Communication overhead is a major factor affecting the performance of collaborative anti-spam systems. We compare the ALPACAS approach with the replicated DCC approach. Fig. 13 indicates the per-test communication cost of both schemes when the number of agents in the system increases from 67 to 600. We conducted experiments with

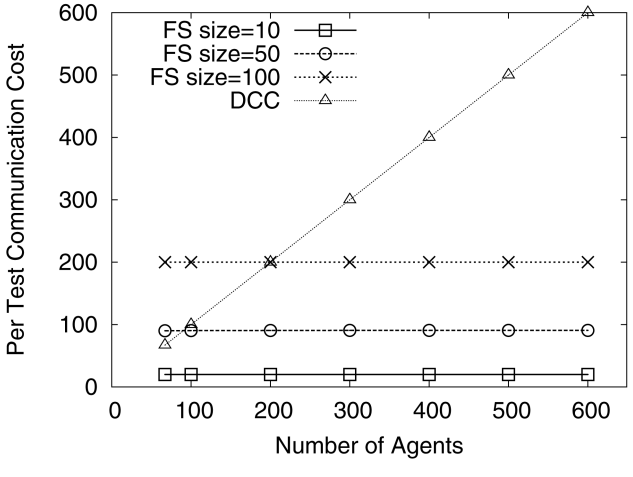


Fig. 13. Communication overheads of the ALPACAS and the DCC.

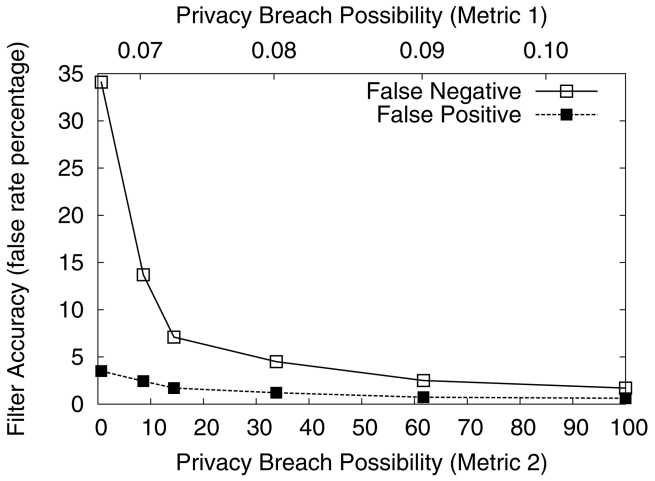


Fig. 14. Privacy and accuracy trade-off.

the size of TFSet being set to 10, 50, and 100. The training phase employed 50 percent of the e-mails in the trace files.

The graph indicates that the per-test communication costs of the DCC approach increases rapidly with increasing number of e-mail agents, whereas the per-test communication costs of the ALPACAS approach essentially remains constant. This result can be explained as follows: In the DCC system, the spam digest database is replicated at each participating agent. Hence, any update to this database has to be reflected at all replicas, which results in high communication overheads. In the ALPACAS approach, the query and publish messages are sent to only the rendezvous nodes of the corresponding e-mails. The number of rendezvous nodes is directly dependent upon the cardinality of the TFSet being employed. Thus, in this scheme, the per-test communication costs depend on the number of FEs in TFSets and not upon the number of participating agents. The results also show that the ALPACAS approach is highly scalable with respect to the number of participating agents.

In our experiments, we used half of the total available data set as training (i.e., 45,925 messages in TREC 2005 data set) and use the other half of the data set for filter accuracy evaluation. In practice, one issue of deploying a

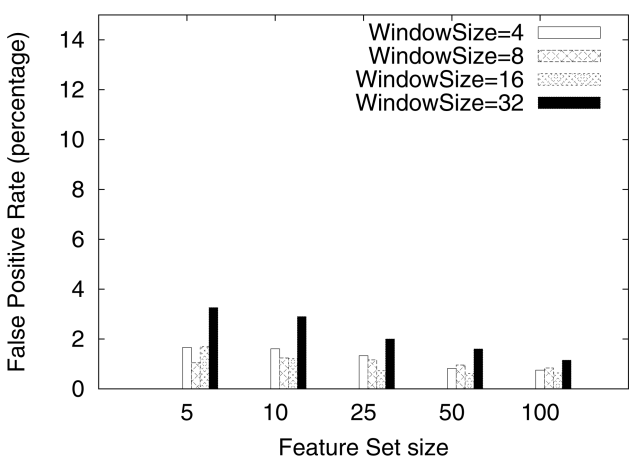


Fig. 15. False positive of ALPACAS for various parameter setup.

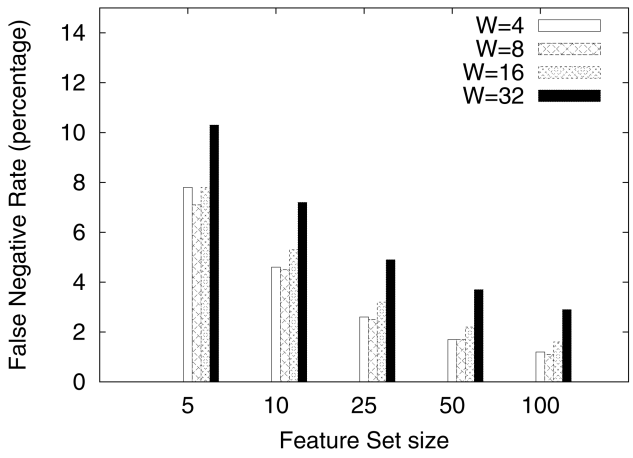


Fig. 16. False negative of ALPACAS for various parameter setup.

system like ALPACAS is the control of data set size at each node. A regular e-mail server usually has a continuous feed of incoming spam and ham messages. The participation needs to balance the trade-off between storing a rich history of spam and ham for classification purpose and controlling the size of data set for the storage and performance constraints. We are currently still investigating the impact of various data set sizes and their freshness on the effect of spam filtering accuracy.

Another performance concern of collaborative systems is the throughput of spam filtration. In general, the filter throughput of collaborative filtering is mostly limited by two factors: the round-trip latency between a node and its collaborative peers and the lookup time for matching FEs. The round-trip time depends on the topology of collaborative systems, but it would not significantly limit the throughput because multiple queries can be processed in parallel by the collaborative system. The bottleneck of performance is often the search time for a closely matched FE at each node. This can be addressed by hash-based classification techniques as proposed in [38].

4.7 Message Transformation Algorithm Analysis

In this set of experiments, we study the effects of various configuration parameters on the effectiveness of the ALPACAS approach. We first study the effects of feature set size and window size on the accuracy of the ALPACAS approach.

Figs. 15 and 16, respectively, show the false positive and the false negative percentages of the ALPACAS approach at various settings of the feature set size and the window size parameters. The results show that employing larger number of FEs yields better classification accuracies. This is because larger feature sets capture more information about the characteristics of individual e-mails. We also observe that ALPACAS approach performs best with medium-sized windows (windows containing 8-10 characters). This observation can be explained as follows: When the window size is very small, the FEs correspond to small, commonly occurring sequences of characters. For example, "agr" can come from either "vi**agr**a" or "**agr**ee." Hence, the feature set of an individual e-mail is likely to exhibit high similarities to both ham and spam e-mails in the knowledge bases,

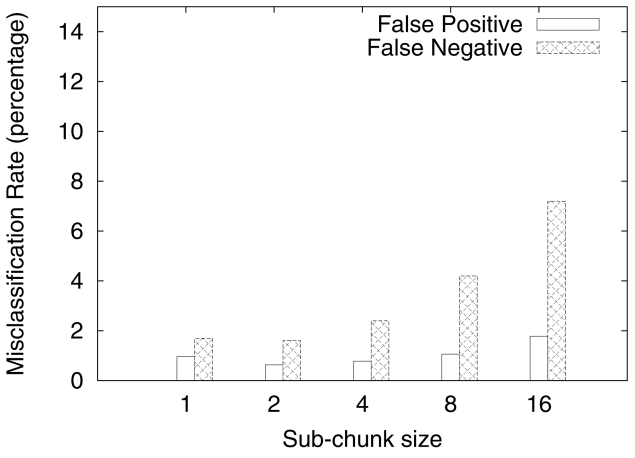


Fig. 17. Effectiveness of controlled shuffling strategy.

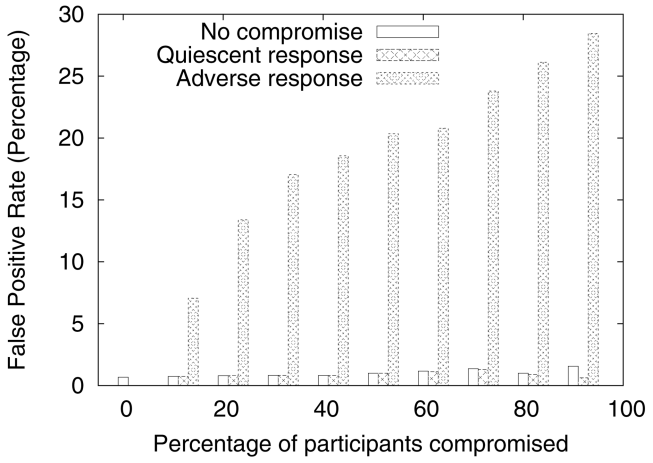


Fig. 18. False positive percentages with compromises.

which affects the classification accuracy. On the contrary, when the window size is set to high values, even *similar* e-mails are likely to have very different feature sets. This is because, when the windows are bigger, each character of the e-mail text appears in several windows. In this scenario, even a few differing characters between two e-mails can affect the similarity of their feature sets to a considerable extent. Thus, when window sizes are very large, feature set of an individual e-mail is likely to have very little similarity to either the spam or the ham e-mails in the knowledge base. This again affects the classification accuracy.

To protect term-level privacy, we propose shuffle method. We assume the entire e-mail is a chunk divided into subchunks by a factor to increase the shuffling degree. Fig. 17 shows the false positive and false negative rates for different subchunk sizes. The results show that when the shuffling degree increases, the accuracy drops. It is because increasing the shuffling degree would break the similarity among e-mails. However, we believe that with a small degree of shuffle, the ALPACAS approach can still achieve a high classification accuracy, and the attackers would spend much more effort to infer the content from a single shuffled FE.

4.8 Resilience to the Compromises

It is possible for an attacker to intrude the ALPACAS system and diminish its spam-filtering capability by deliberately sending deceitful responses to queries or by simply not responding to queries at all. It is necessary that ALPACAS sustains resiliently to these attacks within a certain range even if part of the participants are compromised.

In this section, we study two scenarios of such attacks: 1) *Quiescent response*, where the intruder compromises participating entities and refuses to answer the queries from peers and 2) *Adverse response*, where the intruder does the compromises and adversely sends the matched records back to the peers (i.e., sends ham records back to the query for spam records or sends spam records to the query for ham records).

In general, ALPACAS resists the attack in the *quiescent response* scenario but not in the *adverse response* scenario. Figs. 18 and 19 show the accuracy results for the ALPACAS approach when various percentages of the participating entities are compromised. Both false positive and false negative rates are measured upon the uncompromised entities and then compared to the scenario where no entity is compromised. This experiment is conducted when spam knowledge is shared across the collaborative network.

In the *quiescent response* scenario, our ALPACAS approach shows nearly the same false negative and false positive rates as the case when no compromise happens, even 80 percent of the participating entities are compromised. We believe it is because the spam resources are well distributed and duplicated according to our DHT-based architecture. Being quiescent does not prevent the ALPACAS approach from answering the query by retrieving spam knowledge from uncompromised entities. We also notice that the false positive is better under *quiescent response* scenario when more than 80 percent of the participating entities are compromised. It is because fewer spam knowledge is returned to the querying entity, which favors the decision of ham if the max-similarity comparison is conducted on dissimilar messages.

In the *adverse response* scenario, the ALPACAS approach still has a nearly identical false negative rate until more than 80 percent of the participating entities are compromised. However, the results show a worse false positive rate, which demonstrates the vulnerability of the ALPACAS

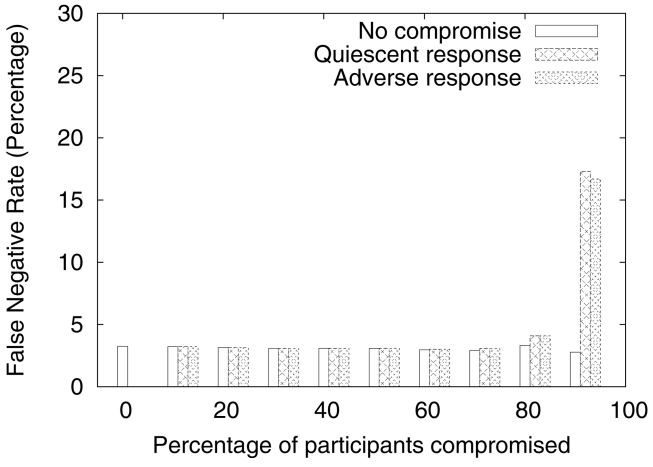


Fig. 19. False negative percentages with compromises.

approach to such attacks. We would like to employ a reputation system to identify such compromised participants in our future work.

5 Discussion

In this section, we discuss some of the important aspects of the ALPACAS approach. We also discuss some of its limitations and possible extensions.

An interesting and important side effect of our approach is that the collaborating agents can discover errors in their own training sets. The training sets for any anti-spam system are a set of e-mails that are manually classified as ham or spam. Due to the human involvement, it is not unusual to find mislabeled e-mails in the training set. Errors in training sets can adversely impact the classification accuracy of any filter. However, in the ALPACAS approach, when an agent tries to upload the feature set of an e-mail into the globally distributed databases, the respective rendezvous agents may notify it of any conflicts they may notice regarding the information being uploaded (such as similar e-mails that may have contradicting labels). This would enable the agent to reverify the e-mail's classification. In the process of our experimental evaluation, we found a small percentage of e-mails in the TREC e-mail corpus that were mislabeled. Specifically, we found 204 pairs of mislabeled e-mails. Each pair contains two very similar messages, one labeled as ham and the other as spam.

In the current design, we use a simple mechanism for the actual message classification. Approaches like statistical filtering [38] can be utilized in conjunction with the feature preservation transformation scheme. One such strategy would be to apply Bayesian filtering on the FEs. We believe that sophisticated classification techniques would further improve the filtering accuracy of the ALPACAS approach. Further, our design of the ALPACAS approach assumes that the e-mail agents are stable (i.e., they have low failure rates). Techniques such as replication and finger-table-based routing [36] can improve the resilience of the ALPACAS approach toward entries and exits of agents.

Collaborative systems like ALPACAS can also benefit from better preprocessing used in existing filters. In our experiments, the complete message bodies of all test messages are used to generate hash values as FEs. We only filtered out those generated from common phrases that occur in both spam and ham message bodies, such as e-mail header fields and HTML tags. In practice, a stronger preprocessing method could be applied to filter out the invisible content, such as HTML comments or camouflage contents that have been identified by other methods.

The current design of the ALPACAS approach assumes that no participating e-mail agent maliciously uploads erroneous information into the knowledge bases. Further, it is also assumed that no e-mail agent in the ALPACAS approach mounts collaborative inference attacks. For example, if the rendezvous agents of an e-mail exchange the FEs they have received as a part of the query message, then they have a better chance of correctly guessing the contents of the e-mail. Preventing these types of malicious behaviors by participating agents is a part of our ongoing work.

6 Conclusion

In this paper, we have presented the design and evaluation of ALPACAS, a privacy-aware collaborative spam filtering framework that provides strong privacy guarantees to the participating e-mail recipients. Our system has two novel features: 1) a feature-preserving transformation technique encodes the important characteristics of the e-mail into a set of hash values such that it is computationally impossible to reverse engineer the original e-mail and 2) a privacy-preserving protocol enables the participating entities to share information about spam/ham messages while protecting them from inference-based privacy breaches. Our initial experiments show that the ALPACAS approach is very effective in filtering spam, has high resilience toward various attacks, and provides strong privacy protection to the participating entities.

Acknowledgments

This work was supported in part by US National Science Foundation (NSF) Grants CNS-0716357 and GRA.GP07.I. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would like to thank the anonymous reviewers for their insightful comments. A shorter version of this paper appeared in the *Proceedings of the 27th Conference on Computer Communications* (*IEEE INFOCOM 2008*) [1].

References

[1] Z. Zhong, L. Ramaswamy, and K. Li, "Alpacas: A Large-Scale Privacy-Aware Collaborative Anti-Spam System," *Proc. IEEE INFOCOM '08,* Apr. 2008.

[2] V. Schryver, *Distributed Checksum Clearinghouse,* http://www.rhyolite.com/anti-spam/dcc/, Nov. 2005.

[3] *Vipul's Razor Anti-Spam System,* Vipul Ved Prakash, http://razor.sourceforge.net/, 2008.

[4] E.S. Raymond, *Bogofilter: A Fast Open Source Bayesian Spam Filters,* http://bogofilter.sourceforge.net/, Nov. 2005.

[5] J. Jung and E. Sit, "An Empirical Study of Spam Traffic and the Use of DNS Black Lists," *Proc. Internet Measurement Conf. (IMC '04),* Oct. 2004.

[6] T. Meyer and B. Whateley, "SpamBayes: Effective Open-Source, Bayesian Based, Email Classifications," *Proc. First Email and Anti-SPAM Conf. (CEAS '04),* July. 2004.

[7] A. Ramachandran and N. Feamster, "Understanding the Network-Level Behavior of Spammers," *Proc. ACM SIGCOMM '06,* Sept. 2006.

[8] B. Leiba, J. Ossher, V.T. Rajan, R. Segal, and M. Wegman, "SMTP Path Analysis," *Proc. Second Email and Anti-SPAM Conf. (CEAS '05),* July 2005.

[9] M.W. Wang, *Sender Authentication: What to Do,* Anti-Abuse Working Group White Paper, http://spf.pobox.com/whitepaper.pdf, 2008.

[10] M. Sergeant, "Internet Level Spam Detection and Spamassassin," *Proc. Spam Conf.,* Jan. 2003.

[11] B. Klimt and Y. Yang, "Introducing the Enron Corpus," *Proc. First Email and Anti-SPAM Conf. (CEAS '04),* July 2004.

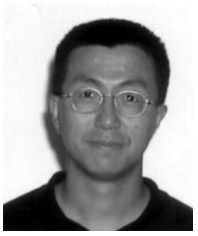
[12] G.L. Wittel and S.F. Wu, "On Attacking Statistical Spam Filters," *Proc. First Email and Anti-SPAM Conf. (CEAS '04),* July 2004.

[13] D. Lowd and C. Meek, "Good Word Attacks on Statistical Spam Filter," *Proc. Second Email and Anti-SPAM Conf. (CEAS '05),* July 2005.

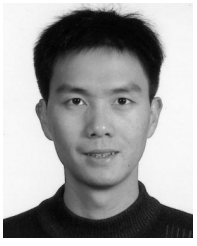
[14] G.V. Cormark and T. Lynam, "Spam Corpus Creation for TREC," *Proc. Second Email and Anti-SPAM Conf. (CEAS),* 2005.

[15] "The Spam Track," *Proc. 15th Text Retrieval Conf. (TREC '06),* Nat'l Inst. of Standards and Technology, http://trec.nist.gov/, 2006.

[16] S. Webb, S. Chitti, and C. Pu, "An Experimental Evaluation of Spam Filter Performance and Robustness against Attack," *Proc. First Int'l Conf. Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom '05),* Dec. 2005.
[17] Cloudmark, *Spamnet Anti-Spam System,* http://www.cloudmark.com/desktop, 2008.
[18] A. Gray and M. Haahr, "Personalised, Collaborative Spam Filtering," *Proc. Second Email and Anti-SPAM Conf. (CEAS),* 2005.
[19] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "P2P-Based Collaborative Spam Detection and Filtering," *Proc. Fourth Int'l Conf. Peer-to-Peer Computing (P2P '04),* http://citeseer.ist.psu.edu/721025.html, Aug. 2004.
[20] F. Zhou and L. Zhuang, *SpamWatch a Peer-to-Peer Spam Filtering System,* http://www.cs.berkeley.edu/~zf/spamwatch, 2003.
[21] J.S. Kong, B.A. Rezaei, N. Sarshar, V.P. Roychowdhury, and P. Boykin, "Collaborative SPAM Filter Using E-Mail Networks," *Computer,* Aug. 2006.
[22] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "Random-Data Perturbation Techniques and Privacy-Preserving Data Mining," *Knowledge Information Systems,* 2005.
[23] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-Diversity: Privacy beyond K-Anonymity," *Proc. 22nd IEEE Int'l Conf. Data Eng. (ICDE '06),* Apr. 2006.
[24] L. Swweney, "K-Anonymity: A Model for Protecting Privacy," *Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, no. 5, 2002.
[25] E. Bertino, B.C. Ooi, Y. Yang, and R.H. Deng, "Privacy and Ownership Preserving of Outsourced Medical Data," *Proc. 21st Int'l Conf. Data Eng. (ICDE),* 2005.
[26] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," *Proc. 16th ACM Conf. Information and Knowledge Management (CIKM),* 2001.
[27] R.J. Bayardo and R. Agrawal, "Data Privacy through Optimal K-Anonymization," *Proc. 21st IEEE Int'l Conf. Data Eng. (ICDE '05),* Apr. 2005.
[28] N. Zhang and W. Zhao, "Distributed Privacy Preserving Information Sharing," *Proc. 31st Int'l Conf. Very Large Data Bases (VLDB),* 2005.
[29] R. Agrawal and R. Srikant, "Privacy-Preserving Data Mining," *Proc. ACM SIGMOD,* 2000.
[30] J. Vaidya and C. Clifton, "Privacy-Preserving Top-K Queries," *Proc. 21st Int'l Conf. Data Eng. (ICDE),* 2005.
[31] A. Broder, "Some Applications of Rabins Fingerprinting Method," *Sequences II: Methods in Communications, Security, and Computer Science.* Springer-Verlag, pp. 143-152, 1993.
[32] Z. Bar-Yossef and S. Rajagopalan, "Template Detection via Data Mining and Its Applications," *Proc. 11th Int'l World Wide Web Conf. (WWW '02),* May 2002.
[33] L. Ramaswamy, A. Iyengar, L. Liu, and F. Douglis, "Automatic Detection of Fragments in Dynamically Generated Web Pages," *Proc. 13th World Wide Web Conf. (WWW '04),* May 2004.
[34] M.O. Rabin, "Fingerprinting by Random Polynomials," technical report, Center for Research in Computing Technology, Harvard Univ., 1981.
[35] R. Tewari, M. Dahlin, H. Vin, and J. Kay, "Beyond Hierarchies: Design Considerations for Distributed Caching on the Internet," *Proc. 19th Int'l Conf. Distributed Computing Systems (ICDCS '99),* May 1999.
[36] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," *Proc. ACM SIGCOMM '01,* Aug. 2001.
[37] L. Ramaswamy, L. Liu, and A. Iyengar, "Cache Clouds: Cooperative Caching of Dynamic Documents in Edge Networks," *Proc. 25th Int'l Conf. Distributed Computing Systems (ICDCS '05),* June 2005.
[38] K. Li and Z. Zhong, "Fast Statistical Spam Filter by Approximate Classifications," *Proc. ACM SIGMETRICS/IFIP Performance,* 2006.



Kang Li received the BS degree in computer science from Tsinghua University, Beijing, in 1995 and the PhD degree in computer science and engineering from the Oregon Graduate Institute of Science and Technology in 2002. He is currently with the Department of Computer Science, University of Georgia. His research interests include computer networks and information security, especially spam resistance and privacy protection. He was the program cochair of the Conference on Email and Anti-Spam (CEAS 2007). He is a member of the IEEE.



Zhenyu Zhong received the BS degree in computer science and application from East China Normal University, Shanghai and the PhD degree in computer science from the University of Georgia, Athens, in 2007. He is currently a research scientist at Secure Computing. His research interests include network security, operating system, and distributed systems. His work especially focuses on large-scale intrusion detection and defense, anti-spam, and privacy protection. He is a student member of the IEEE.

Lakshmish Ramaswamy received the PhD degree in computer science from Georgia Institute of Technology in 2005. He is currently an assistant professor in the Department of Computer Science, University of Georgia. His research interests are broadly in the area of distributed systems and more specifically in performance, scalability, security and privacy of Internet services, overlay networks, events-based middleware, and mobile systems. He is the recipient of the Best Paper Award at the 13th World Wide Web Conference (WWW 2004) and the 2005 Pat Goldberg Best Paper Award. He has served on the program committees of several international conferences and workshops. He was the program cochair of the First International Workshop on Distributed Event Processing, Systems and Applications (DEPSA 2007) and the International Workshop on Semantic Enabled Networks and Services (SENS 2006). He is a member of the IEEE and the IEEE Computer Society.