# Improving Digest-Based Collaborative Spam Detection

**Slavisa Sarafijanovic**
EPFL, Switzerland
slavisa.sarafijanovic@epfl.ch

**Sabrina Perez**
EPFL, Switzerland
sabrina.perez@epfl.ch

**Jean-Yves Le Boudec**
EPFL, Switzerland
jean-yves.leboudec@epfl.ch

## ABSTRACT

Spam is usually sent in bulk. A bulk mailing consists of many copies of the same original spam message, each sent to a different recipient. The copies are usually obfuscated, i.e. modified a bit in order to look different from each other. In collaborative spam filtering it is important to determine which emails belong to the same bulk. This allows, after observing an initial portion of a bulk, for the bulkiness scores to be assigned to the remaining emails from the same bulk. This also allows the individual evidence of spamminess to be joined, if such evidence is generated by collaborating filters or users for some of the emails from an initial portion of the bulk. Then, the observed bulkiness and the estimated spamminess of a bulk can be used to better filter the remaining emails from the same bulk.

The work by Damiani et al. [2] ("open-digest paper") is well know and often cited for its positive findings about the properties of a digest-based collaborative spam detection technique. The technique produces similar digests out of similar emails, and uses them to find out which emails belong to the same bulk. Based on the experimental evaluation, the paper suggests that the technique provides bulk-spam detection that is robust to increased obfuscation efforts by spammers, and low miss-detection of good emails.

We first repeat and extend some of the open-digest paper [2] experiments, using the simplest spammer model from that paper. We find that the conclusions of the open-digest paper are rather miss-leading. Then we propose and evaluate, under the same spammer model, a modified version of the original digest technique. The modified version greatly improves the resistance of spam detection against increased obfuscation effort by spammers, while keeping miss-detection of good emails at a similar level. Based on the observed results, we discuss possible additional modifications and algorithms that could be added on top of the modified digest technique to further improve its filtering performance.

## Keywords

Email, spam, open digest, similarity hashing, data representation, collaborative, detection, filtering, obfuscation, robustness.

*In Proceedings of the 2008 MIT Spam Conference,* Cambridge, Massachusetts, USA, March 27-28, 2008.

## 1. INTRODUCTION

### 1.1 Background on Collaborative Spam Detection Using Similarity Digests

An important feature of spam, which can be exploited for detecting it easier, is its bulkiness. A spam bulk mailing consists of many copies of the same original spam message, each sent to a different recipient or group of recipients. The different copies from the same bulk are usually obfuscated, i.e. modified a bit in order to look different from each other. Spammers apply obfuscation in order to make collaborative spam detection more difficult.

Indeed, in collaborative spam detection it is important to have a good technique for determining which emails belong to the same bulk. This allows, after observing an initial portion of a bulk, for the bulkiness scores to be assigned to the remaining emails from the same bulk. If the collaborative spam detection is based purely on the evaluation of bulkiness, each recipient must be equipped with a white lists of all the bulky sources from which she or he wants to receive emails.

Having a good technique for determining which emails belong to the same bulk also allows for the individual evidence of spamminess to be joined, if such evidence is generated by collaborating filters or users for some of the emails from an initial portion of the bulk. The observed bulkiness and the estimated spamminess of a bulk can then be used to better filter the remaining emails from the same bulk. Collecting and using the evidence of spamminess is especially useful if the reputation of spam reporters is evaluated and used, in which case the collaborative detection may be relatively safe to use even if the recipients are not equipped with white lists of the bulky sources from which they want to receive emails.

A good source of the evidence of spamminess, which is increasingly used in practice, are the emails tagged as spam by those users that have and use a "delete-as-spam" button in their email-reading program. Automated and probabilistic tagging is also possible, e.g. by use of Bayesian filters' scores, or by use of "honey pot" email accounts that are not associated to real users but only serve to attract unsolicited bulk emails.

#### *1.1.1 Existing digest-based approaches*

A well-known technique for detecting whether emails belong to the same spam bulk is presented and evaluated in the "OD-paper" by Damiani et al. [2] (OD stands for Open Digest). OD-paper is often cited in the literature related to digest-based collaborative spam filtering, as it gives very positive results and conclusions on the resistance of the tech-

nique to the increased obfuscation-effort by spammers. It also shows that the technique is expected to have very low false-positives.

The technique produces similar digests out of similar emails, and uses them to find out which emails belong to the same bulk. The digests are produced from the complete email or from the complete predefined parts of the email. The digest queries are submitted to a global database, and the replies indicate the number of similar messages (queries) observed by the database. The technique is further explained in 1.1.2. It is important to mention that such a technique is implemented by DCC [3], and that the DCC database of digests is used by SpamAssassin [7] (a very popular open-source antispam software integrated in many spam filters).

The peer-to-peer system for collaborative spam filtering by Zhou et al. [9] is another well-known and often cited digest-based antispam technique. It uses multiple digests per email, created from the strings of fixed length, sampled at random email positions. They apply however the exact matching instead of a similarity matching between the digests, as required by the rest of their system to work. Even modest spam obfuscation is able to alter some of the bits of such generated digests, which prevents their system from detecting spam bulks. Their analysis results in a different conclusion, because they use rather unrealistic obfuscation (which alters the created digests with a very small probability) to test their solution.

The system proposed by Sarafijanovic and Le Boudec [6] produces multiple digests per email, from the strings of fixed length, sampled at random email positions, and it uses similarity matching. Additionally, it uses artificial immune system algorithms to process the digests before and after exchanging them with other collaborating systems, in order to control which digests will be activated and used for filtering of the incoming emails. The system shows good performances in detecting spam bulk under a specific spammer model, but an additional evaluation is needed for more general conclusions about its abilities. As the factorial analysis is missing, it is not clear whether the observed good performances are due to the way the digests are produced (e.g. as compared to the standard digest from the OD-paper [2]), or due to the advanced algorithms used by the system.

The direct comparison of the above explained different ways of producing the digests from emails, according to our best knowledge, has not yet been scientifically evaluated.

### 1.1.2 Open-digest technique from OD-paper [1]

**Data representation: Digests produced using Nilsimsa hashing.** The open-digest technique from the OD-paper represents an email by a 256-bits digest. The transformation is performed using Nilsimsa hashing [5]. This is a locally sensitive hash function, in sense that small changes in the original document may impact only few bits of the digest. That means that similar documents will have similar digests, in sense of a small Hamming distance between them. With the standard hash functions small changes in the original document usually result in a digest that is completely different from the digest of the original document.

OD-paper gives a detailed description of the Nilsimsa hashing. In summary, a short sliding window is applied through the email. For each position of the window, the trigrams from the window are identified that consist of the letters from the predefined window positions (that are close to each other, but not only consecutive-letters trigrams are used). The collected trigrams are transformed, using a standard hash, to the positions between 1 and 256, and the accumulators at the corresponding positions are incremented. Finally, the accumulators are compared to the mean or to the median of all the accumulators, and the bits of the digest are set to 0 or 1, depending on whether the corresponding accumulators are bellow or above the threshold.

The digests are called "open" because: a) the digests computation method is assumed to be publicly known; b) the used similarity hashing hides original email text, so the privacy of the content is preserved even if the digests are openly exchanged for collaborative filtering.

**Detection algorithm: Counting similar digests.** OD-paper imagines digest-based spam detection by use of digest-queries to a central database that contains the digests of the emails recently observed by the collaborating spam filters (the previous queries)[1], and counting the number of the similar digests (emails) found in the database.

## 1.2 Our Work and Contributions

### 1.2.1 Re-evaluation of the open-digest technique from OD-paper

We first repeat and then extend some of the open-digest paper [2] experiments, using the simplest spammer model from that paper. More precisely we re-consider the experiments with spammer which obfuscates emails by addition of random characters. We find that some of the most important conclusions of the open-digest paper are rather missleading.

### 1.2.2 Proposal and evaluation of an alternative to the open-digest technique from OD-paper

We propose and evaluate a modified version of the original digest technique. The modified technique uses the same Nilsimsa hashing function, but instead of producing one digest from the complete email, it produces multiple digests per email, from the strings of fixed length, sampled at random email positions. Basically, we only change the way of producing the digests from emails. For fairness of the comparison, we evaluate both techniques while having in mind the same simple detection algorithm that was the basis for the original OD-paper experiments.

We show that the modified technique greatly improves the resistance of spam detection against increased obfuscation effort by spammers, while keeping miss-detection of good emails at a similar level. Based on the observed results, we discuss possible additional modifications and algorithms that could be added on top of the modified digest technique to further improve its filtering performance.

## 1.3 Organization of The Paper

In the next section (Section 2) we recreate and extend some of the OD-paper experiments, discuss the results of these experiments, and revise the conclusions of the OD-paper. In Section 3, we propose and evaluate a modified version of the original digest technique, and compare it to the original technique from the OD-paper. In Section 4 we summarize the paper results and achievements. Based on the observed results, we outline possible directions for further improvements in digest-based spam detection.

[1]We conclude this from the description of the OD-paper experiment for evaluation of false positives, in which they compare digests of good emails to the digests of both spam and good emails.

## 2. REVISITING RESULTS AND CONCLUSIONS OF OD-PAPER

As mentioned in Section 1.1.2, OD-paper assumes use of a database of digests from ham and spam emails, e.g. created out of those emails that are observed recently in the emailing network. It compares the emails to be filtered to the emails from the database. As good emails are unrelated to each other and to spam emails, their digests are expected to match to the digest from the database with a small probability. On the other side, the digests from spam emails should with a high probability match the digests in the database that come from the same spam bulk. Spam digests are also expected to not match many of the digests that come from other emails and other bulks. Therefore, the evaluation metrics must be slightly differently computed for evaluating the detection of bulky spam emails then for evaluating the miss-detection of good emails. Some possible evaluation metrics and the used evaluation metrics are discussed in Section 2.2. The performed experiments and the computed evaluation metrics are detailed in Sections 2.3-2.5.

### 2.1 Considered Spammer Model

OD-paper evaluates the spam detection technique explained in Section 1.1.2 against few spammer attacks: addition of random characters, aimed addition of characters that takes into account the details of how the digests are produced, replacement of words by synonyms, and perceptive substitution of characters.

In this paper we do all evaluations using the first spammer model (addition of random characters).

### 2.2 Metrics Used to Evaluate The Open-Digest Technique From The OD-paper

To assess the ability of spam bulk detection and good email miss-detection, as the first option, one could simulate the real scenario of submitting the digests of the emails to be filtered to a database, and receiving back the counters of how many digests in the database is matched by the submitted digests. And then comparing the counters to their ideal values (0 - for ham emails; number of earlier emails from the same bulk - for spam emails).

The second option would be to do email-to-email comparisons and estimate the probabilities of matching between unrelated emails (e.g. between a ham email and emails from the database), and between related emails (spam emails from the same bulk). Then, knowing a possible size of the digest database, one could calculate the probabilities of the counter values returned by the database upon a digest-query.

The OD-paper uses the second option for estimating false detection of good emails. It also uses the second option to evaluate detection of spam bulk, but instead of showing the probabilities of email-to-email matching it shows the average of the *Nilsimsa Compare Values*[2] between the compared digests.

Computing the average NCV for spam emails makes sense only if the comparisons to the spam emails from the same bulk are used. Actually, the OD-paper experiments for evaluating detection of spam bulk are done exactly that way.

In addition to repeating the OD-paper experiment that evaluates spam bulk detection in the same way as it is done originally in OD-paper, we also evaluate spam detection in the case when the queried database contains both spam and good emails, in order to make the evaluation more realistic and complete. In the mixed database case it doesn't make sense to use as a spam bulk detection metric the average of the average NCVs each spam email scores to the emails from the database (bulk detection NCV results would be polluted by NCV results between unrelated emails). Instead, we compute the average of the *maximum NCVs* each spam email scores against the emails of the database (as a simple and logical alternative). In parallel, we also evaluate *email-to-email matching probabilities*, and we also show the *histogram of email-to-email NCVs* (for these two metrics, in the case of spam-detection evaluation, we account only for comparisons against the emails from the same bulk - otherwise bulk-matching results would be masked by unrelated-emails-matching results). Another possibility would be, for example, to count the number of the NCV scores above a given threshold - in which case it would make sense to account for the comparisons against all the emails from the database (similar to the "maximum NCV" case).

As it is shown in the rest of the paper, use of adequate metrics is important for a correct and reliable assessing of the properties of the considered spam bulk detection technique.

### 2.3 "Spam bulk detection" experiment ($SPAM - SPAM\_BULK$)

This section gives the details of the experiment in which we reproduce the result from the Figure 2 of OD-paper (spam bulk detection under the"random addition" attack).

The $SPAM - SPAM\_BULK$ experiment:

- 20 emails are sampled randomly from the used spam repository (we use 20030228_spam_2.tar.bz2 spam repository[3] from the Spamassassin public corpus [8]);
- a pair of obfuscated copies is created from each of the 20 emails - random characters are added to the end of each email in the amounts being a percentage (we call this parameter obfuscation ratio) of the original emails size;
- the digests of the two copies from a pair are compared to each other by computing the NCV (Nilsimsa Compare Value[4]);
- mean and 95% confidence are calculated out of the 20 NCVs;
- the above steps are repeated, except the first one, for other values of the ratio of added characters (using also the values larger from those evaluated in the OD-paper).

The results of this experiment are shown in Figure 1. The source code and the used email databases of this and other experiments from this paper are made available online [1].

[2]**Nilsimsa Compare Value (NCV)** between two digests is equal to the number of the equal bits at the same positions in the two digests, minus 128 (for the digests of 256 bits). The higher NCV indicates the higher similarity of the texts from which the digest are computed (which can be understood from Section 1.1.2).

[3]OD-paper used the spam repository from SpamArchive (www.spamarchive.org) which is not any more available on the Internet. We also were not able to obtain it from the authors of OD-paper. Thus we decided to use a SpamAssassin repository.

[4]See footnote 2

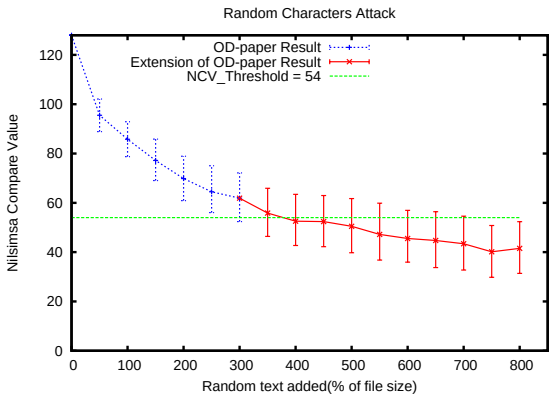



**Figure 1: Repeated and extended OD-paper result for bulk spam detection under the "adding random text" spammer model. Repeated OD-paper experiment (dotted blue line) recreates pretty well the curve from the Figure 2 of the OD-paper. Extension of the same experiment (solid red line) for higher values of the percent of the added random text (the percent of non-obfuscated emails text size - we call this parameter obfuscation ratio) indicates that the open-digest technique actually is vulnerable to random text additions - opposite to the conclusion from the OD-paper.**

### 2.3.1 *Parameters and results discussion*

**NCV comparision threshold.** The used NCV comparison threshold is the same as in the OD-paper(the dashed green line of Figure 1). The OD-paper authors calculate and suggest the value 54 as the value that should ensure low miss-detection of good emails (this is discussed more in detail within the experiment dedicated to the evaluation of the miss-detection of good emails, Section 2.5).

**Recovered OD-paper experiment.** The dotted blue line fits very well the result from the Figure 2 of OD-paper, which suggests that we recreated the OD-paper experiment properly[5] (i.e. the NCVs in this experiment are "spam to spam from the same bulk", as we assumed).

**Extended experiment: different conclusions.** Based on the observation that the average NCV is above the matching threshold for the obfuscation ratios they tested, the authors of OD-paper conclude that the bulk spam detection is well resistant against increased obfuscation efforts by spammers. Our extension of the experiment (solid red line) shows that using even slightly higher obfuscation ratios then those tested in OD-paper brings the average NCV bellow the threshold, which invalidates the conclusion of OD-paper, as this small additional obfuscation effort is easy for spammers to perform.

**NCV metric usability.** However, though we agree that the average NCV gives some indication about the resistance of the detection to the increased obfuscation efforts by spammers, we suggest and show in the following experiments and figures that the use of additional metrics such are probability of email-to-email matching (on the level of the digests) and the histogram of NCVs allows us to much better see the qualitative and the quantitative impact of the obfuscation to the detection of spam and miss-detection of good emails.

[5]The OD-paper does not specify this experiment in detail, and we were not able to obtain the original code of the OD-paper experiments from the web or from the OD-paper authors.

## 2.4 "Spam bulk detection" experiment ($SPAM - DB$)

This section gives the details of the experiment in which we compare digests of spam emails against a database of digests of both spam and good emails. Such database of digests is used in the experiment done in OD-paper for evaluating false-positives (comparison of good emails to the database of digests). This corresponds to the realistic scenario in which the database of digest is made out of the previous queries to it. However, in the "spam bulk detection" experiment ($SPAM - DB$), we still can define and evaluate some metrics that consider only the comparisons of spam emails to the emails from the same bulk, and we do so.

The $SPAM - DB$ experiment:

- a 100-emails database (DB set) is created by sampling randomly 50 spam emails from 20030228_spam_2.tar.bz2 SpamAssassin's spam repository[6] and 50 ham emails from 20021010_easy_ham.tar.bz2 SpamAssassin's ham repository[7]; whenever an email is sampled, it is removed from the original database;
- the 50 spam emails from the DB set are copied into another set (set of spam "to be checked", i.e. the SPAM set);
- the 100 spam emails from the DB and SPAM sets are obfuscated by adding random characters to the end of each email in the amounts being a percentage (we call this parameter obfuscation ratio) of the original emails size;
- the 100-emails database DB is converted into a 100-digest database (DB' set), by producing a digest from each email;
- for each of the 50 emails from the SPAM set the following computations are performed: a) the digest of the email is compared to all the digests from the DB' set, the NCV is computed for each comparison, and the maximum NCV is computed over the comparisons; b) the NCVs obtained for the comparisons to the spam emails that origin from the same original spam email (i.e. belong to the same spam bulk) are compared to the threshold (54), and the number of the matched emails (when $NCV >= 54$ there is a matching) is divided with the number (lets call it n) of the emails in DB that belong to the same bulk, in order to obtain an estimate of the probability for that email to match other spam email from the same bulk (in our case the estimated probability will be a binary number, because n is equal to 1, due to the experiment design);
- mean and 95% confidence interval are calculated out of the 50 results obtained for the SPAM emails (for each spam from SPAM set we obtained a maximum NCV and an indicator (0 or 1) of matching to another email from the same bulk);
- the above steps are repeated, excluding steps 1 and 2, for other values of the ratio of added characters.

[6]See footnote 3

[7]OD-paper used the legitimate messages from comp.risks (www.usenet.org) which are not any more available. We also were not able to obtain them from the authors of OD-paper. Thus we decided to use a SpamAssassin repository.

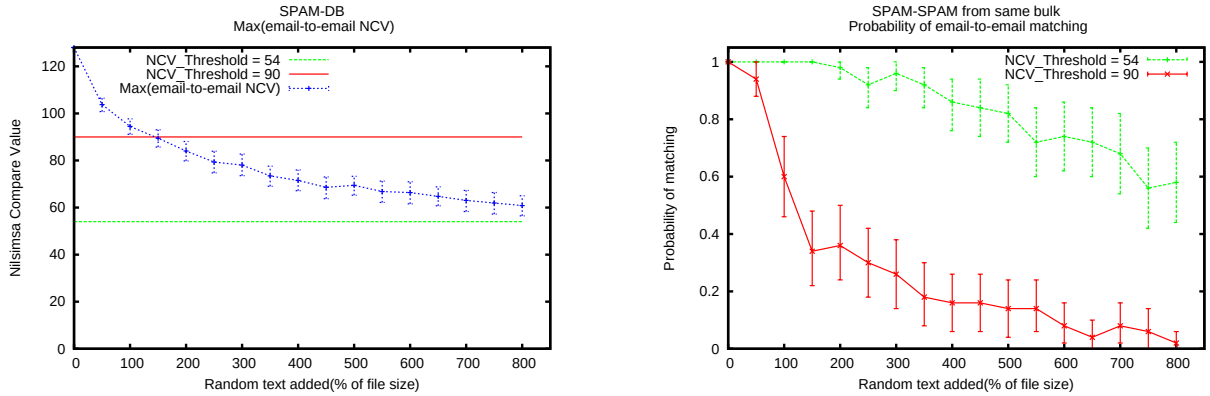



**Figure 2: OD-paper digest technique: Bulk spam detection under the "adding random text" spammer model. The case in which the database DB of digests contains both spam and ham. We see the technique is vulnerable to the increased effort by spammer, but the impact of the increased obfuscation can not be correctly concluded from simple extension of the OD-paper experiments and by looking at only the averaged NCVs. The probability of email-to-email matching is much more informative and allows for a more proper qualitative and a more detailed quantitative assessing of the performances of the detection scheme. The observed probability of email-to-email results suggest that the increased obfuscation will substantially increase the number of emails from the bulk that pass the users' filters before the critical number of the digest from the bulk is collected in the digests database, especially for the NCV compare limit values that are noticeably higher then 54 (e.g. for 90). It is shown in the next experiment (Section 2.5) that for a low miss-detection of good emails NCV compare limit value must be much higher then 54.**

### 2.4.1 *Parameters and results discussion*

The obtained mean and CI for the maximum NCV are shown, in function of the obfuscation ratio, in Figure 2(a). The (estimated) probability of matching between a spam email from SPAM set - and an email from DB that is from the same bulk - is shown in Figure 2(b) (mean and CI, in function of the obfuscation ratio), for the two values of the matching threshold. The histogram of spam to spam from the same bulk NCVs is shown in Figure 3(a) (for non obfuscated spam emails) and in Figures 3(b)-3(d) (for slightly, moderately and heavily obfuscated spam emails).

**Max(email-to-email NCV) metric.** It should first be well understood that the the means and CIs of the maximum NCVs (Figure 2(a)) that the spam-email digests scored against the digests from the database is practically a very similar metric to the means and CIs of NCVs shown in Figure 1 (the recreated and extended OD-paper result). We recall that in the $SPAM-SPAM\_BULK$ experiment (Figure 1) the spam-email digests are compared only to the digests from the same spam bulk, and the mean observed values are shown in the figure. Here, in the $SPAM-DB$ experiment (Figure 2(a)), the spam-email digests are compared to the digest from the database, and the database digests come from good emails, unrelated spam emails, and spam emails from the same bulk. The $Max()$ operator will in most cases select those NCVs that spam emails scored against spam emails from the same bulk, because the unrelated emails usually have much lower mutual NCVs.

However, we can notice that the NCV curve on the Figure 2(a) has higher values than the NCV curve on the Figure 1. The difference is especially evident for higher obfuscation ratios. The reason for this difference is a relatively wide distribution of NCVs between unrelated emails (which can be seen e.g. from the Section 2.5 experiment, Figure 5). So, for high obfuscation ratios, for which the mean NCV between spam emails from the same bulk becomes low, there is a high chance that some spam emails will score a higher NCV value against one of many unrelated emails from the database (then that NCV value gets picked up by the $Max()$ operator), then against just one considered obfuscated copy from the same bulk. This suggests use of higher NCV threshold values, in order to eliminate or minimize the effect of the mentioned "wide NCV distribution" phenomena. The mentioned "wide NCV distribution" phenomena is especially relevant for miss-detection of good emails and is additionally discussed in Section 2.5.

**NCV threshold values.** All the metrics are computed for the two values of the NCV threshold, 54 and 90. The NCV threshold 54 allows comparison of the results and conclusions with those from OD-paper. We consider one additional threshold value (90) in order to illustrate the effect of the threshold change on both detection of spam (this experiment) and miss-detection of good emails (the experiment of Section 2.5). We pickup the second value to be much higher then 54, because in the next experiment (Section 2.5) we find that for the NCV threshold value 54 the miss-detection of good emails is very high. In this paper we do not optimize the NCV threshold value. This could be done by applying the standard procedure for optimizing parameters of spam filters, i.e. by finding the NCV threshold value that achieves an appropriate compromise between the miss-detection of good emails and non-detection of spam.

**Vulnerability to obfuscation: NCV versus probability of email-to-email matching results.** If we try to even only qualitatively conclude about the bulk spam detection vulnerability to the obfuscation, by looking only at the means and CIs of the maximum NCVs that spam emails scored against the emails from the database, we can

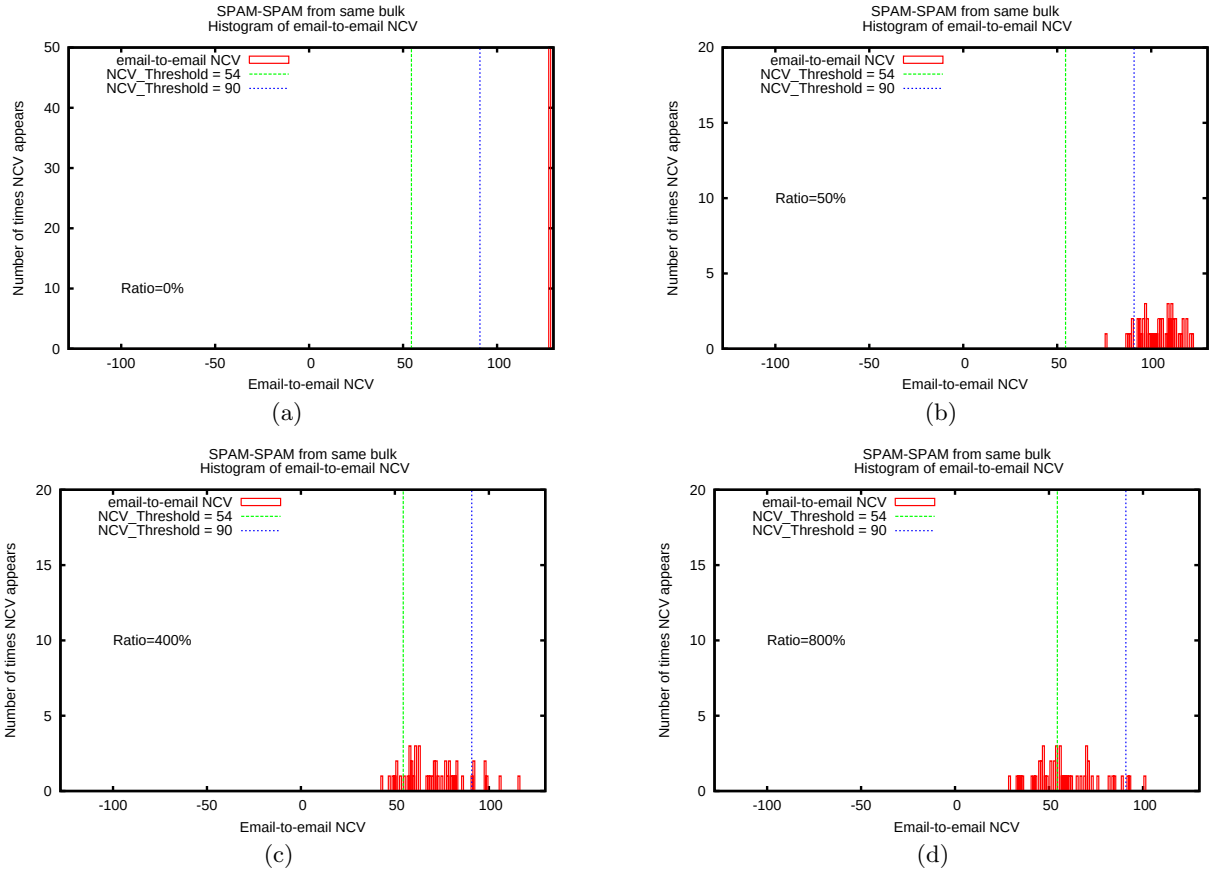



**Figure 3: OD-paper digest technique: Impact of increased obfuscation by spammer on mutual matching of emails from the same bulk.** **The increased ratio of added random text renders most of the digest to become not useful for spam bulk detection. NCV threshold value must be high to the low miss-detection of good emails constraint (as shown in Section 2.5).**

see (Figure 2(a)) that for the NCV threshold 54 the CIs are always above the threshold . We could conclude that the detection is well resistant to the obfuscation. For the NCV threshold 90, the complete CI is above the threshold for the obfuscation ratios up to 100%, but the complete CI is bellow the threshold already at the obfuscation ratio 200%, and stays bellow the threshold for the higher obfuscation ratios. Following the reasoning used in OD-paper, we could (wrongly) conclude that, for the NCV threshold 90, the detection is well resistant to the obfuscation ratios up to 100%, and that it is not resistant for the obfuscation ratios above 200%.

However, if we look at the estimated probability of email to email from the same bulk matching (Figure 2), we can have more correct qualitative and even very good quantitative conclusions about the detection efficiency and resistance to the obfuscation. For the NCV threshold 90, the probability of matching is effectively rather similar for the obfuscation rations 100% and 200%, in the sense that in both cases only few digests from a bulk in the database would ensure high probability that at least one, and actually a large portion of them of them, match the new digests from the same bulk. From the probability of email to email detection results it is possible to compute the number of the the digest from a spam bulk that have to be collected in the database in order to achieve a specified high probability for the new digests from the same bulk to match a specified number of the digests from the database (requiring more than one match may be needed in order to achieve low miss-detection of good emails).

Already visually and without detailed computation we can see that stronger obfuscation will not completely prevent detection of spam, but will substantially increase the number of spam emails from a bulk that will bypass the detection.

**Obfuscation under x-ray: NCV histogram.** Figures 3(a)-3(d) show visually what happens with the digests under the obfuscation of spam emails. Strong obfuscation renders most of the digests from a bulk to become not-useful for matching other digests from the same bulk (the NCVs bellow the threshold). One should be aware that only high NCV threshold values are acceptable for practical use, as required for low miss-detection of good emails (as shown in Section 2.5).

## 2.5 "Good emails miss-detection" experiment ($HAM - DB$)

This section gives the details of the experiment in which we test the digests of good emails against a database of the digests of both spam and good emails. The same experiment is done in OD-paper for evaluating false-positives. Only the

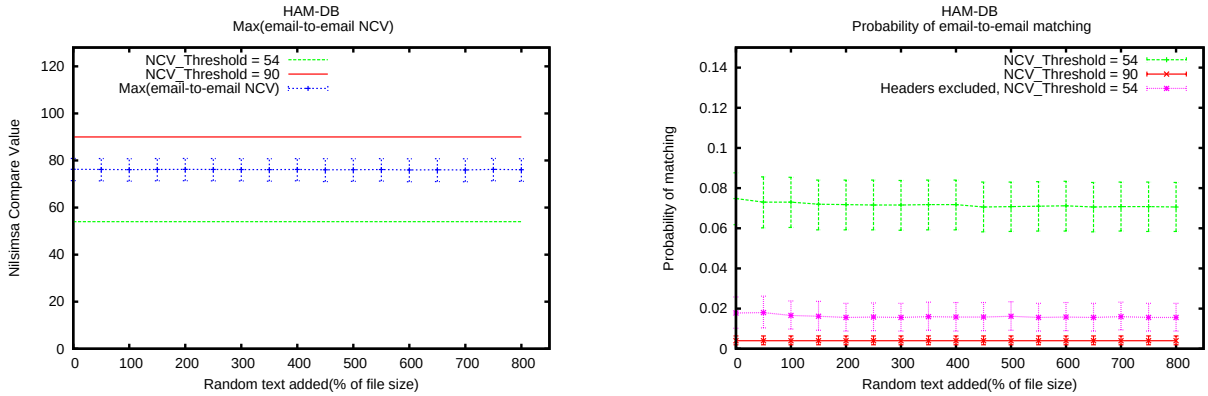



**Figure 4: OD-paper digest technique: Miss-detection of good emails under the "adding random text" spammer model.** **We consider the case in which the database DB of digests contains both spam and ham digests (the same case is considered in the OD-paper). We can see that for TH=54 (the only case evaluated in OD-paper) miss-detection is much higher then advocated (and partially experimentally supported) in OD-paper.**

used email repositories are different, and the number of the compared emails is different, which should normally not affect the results (due to the experiment design), except for one of the used metrics that is dependant on the number of compared emails ($Max(email-to-email\ NCV)$ metric).

The $HAM-DB$ experiment:

- a 100-emails database (DB set) is created by sampling randomly 50 spam emails from 20030228_spam_2.tar.bz2 SpamAssassin's spam repository[8] and 50 ham emails from 20021010_easy_ham.tar.bz2 SpamAssassin's ham repository[9]; whenever an email is sampled, it is removed from the repository;
- the 50 spam emails from the DB set are obfuscated by adding random characters to the end of each email in the amounts being a percentage (we call this parameter obfuscation ratio) of the original emails size;
- the 100-emails database DB is converted into a 100-digests database (DB' set), by producing a digest from each email;
- another 50 hams (hams "to be checked", i.e. the HAM set) are sampled from the ham repository (whenever an email is sampled, it is removed from the repository);
- for each of the 50 emails from the HAM set the following computations are performed: a) the digest of the email is compared to all the digests from DB', the NCV is computed for each comparison, and the maximum NCV is computed over the comparisons; b) each NCV is compared to the threshold (54), and the number of the matched emails (when $NCV >= 54$ there is a matching) is divided with 100 in order to obtain an estimate of the probability for that email to match other unrelated ham and spam emails;
- mean and 95% confidence interval are calculated out of the 50 results obtained for the HAM emails (for each ham from HAM set we obtained a maximum NCV and a probability of matching emails from DB);
- the above steps are repeated, excluding steps 1 and 4, for other values of the ratio of added characters.

[8]See footnote 3
[9]See footnote 7

### 2.5.1 *Parameters and results discussion*

The obtained mean and CI for the maximum NCV are shown, in function of the obfuscation ratio, in Figure 4(a). The probability of matching between a good email from HAM set and an email in the DB database of spam and good emails is shown in Figure 4(b) (mean and CI, in function of the obfuscation ratio), for the two values of the matching threshold. The histogram of email-to-email NCVs is shown in Figure 5(a) (for non obfuscated spam emails in DB) and in Figure 5(b) (for heavily obfuscated spam emails in DB).

**Miss-detection of good emails is much higher then advocated in OD-paper.** Though we implement the same experiment as the one described in OD-paper[10,11] (and detailed here in Section 2.5), we do not obtain the zero false positives (miss-detection of good emails) result of the OD-paper. Only the used email repositories and the number of mutually compared digests in our experiment are different then in the OD-paper experiment, but that should normally not have big effect on the results of the experiment. Even if the miss-detection probability would be very small, as OD-paper experiment uses higher number of digest comparisons it should easier discover the cases of miss-detection. However OD-paper states that such cases were not observed in their experiment, though in our experiment the observed ratio of matching between a good email digest and a digest from the DB database is around 2% if the headers are ex-

[10]OD-paper does not specify whether the digests are produced from the complete emails (including headers) or only from the contents of the emails. Therefore we perform the experiment both with and without exclusion of headers, for the same NCV threshold value (54) as the one used in OD-paper, but in both cases the miss-detection of good emails is non-zero, contrary to the finding of OD-paper.

[11]As elimination of headers or any other preprocessing of the emails before computing the digests is not mentioned in OD-paper, and as the headers are usually present in the the databases used for evaluation of emails, in the remaining experiments we compute the digests from the complete emails. We were not able to obtain the original OD-paper code and email repositories (see footnotes 3, 5, and 7). However, the well recovered curve from Figure 2 of OD-paper (our Figure 1) suggests that the digests of OD-paper are also computed from the complete emails.

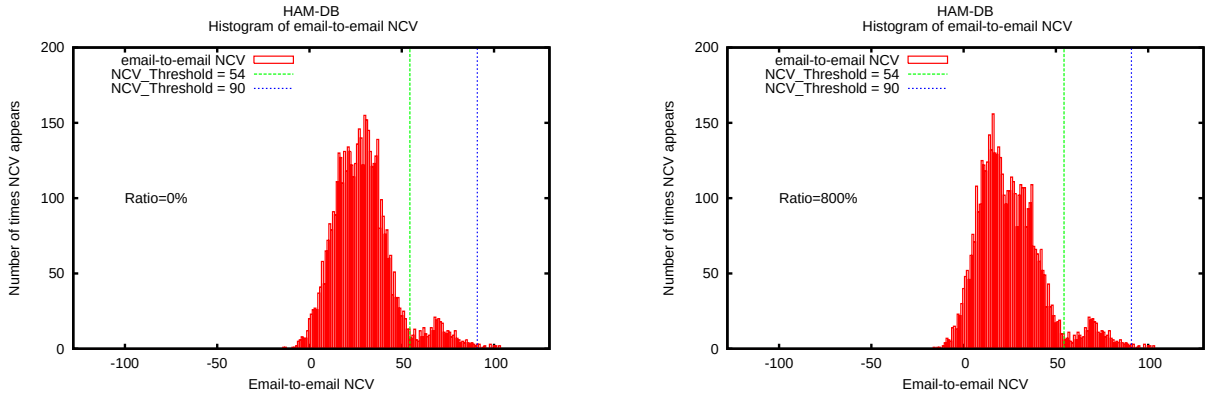



**Figure 5: OD-paper digest technique: Impact of the NCV threshold on the (undesirable) matching of good-email digests to the database of digest from both spam and good emails.** **We can see that the NCV threshold should be set very high in order to provide low miss-detection of good emails, much higher then the value TH=54 suggested by OD-paper for good detection of bulk spam. However, very high values of the NCV threshold are not possible to use as they increase the vulnerability of spam detection to the increased obfuscation by spammer (as shown in Section 2.4, Figure 2). We can also see that the obfuscation of spam has no effect on the miss-detection of good emails.**

cluded when producing the digests (the doted pink line on the Figure 4(b)), and it is about 7.5% if the digests are produced from the complete emails (the dashed green line on the Figure 4(b)), with the same NCV threshold value (54) as in the corresponding OD-paper experiment.

A possible reason for the different results could be a bug in one (or both) of the experiment implementations. However we observe the scheme under multiple metrics and find all of them consistent and logical within each experiment, as well as when we compare them over the different experiments.

Another possible reason could be that one of the experiments uses the non-representative repository of good emails. However, among two real repositories, the one that shows that an email filtering technique might produce good emails miss-detection (with a probability that is rather high) is more relevant: An email filtering technique is rather not usable if it produces (high) miss-detection of good emails in some cases (testing repositories), even if there are cases (testing repositories) in which it does not produce miss-detection of good emails (unless it is possible to identify classes of users that correspond to the "good"-case ham repositories, so that at least these users may use it safely).

**Possibility to decrease miss-detection of good emails is very limited.** Similar as in the previous experiments, from the NCV means and CIs (Figure 4(a)), we cannot tell a lot about the effectiveness of the filtering technique or about about the effect of the NCV threshold change on this effectiveness. Actually, in this experiment, from Figure 4(a) we can at least conclude that the obfuscation does not impact the miss-detection of good emails.

From the Figure 4(b) we can see that changing the NCV threshold from 54 to 90 decreases the observed ratio of unwanted matching between the digests almost by an order of magnitude. But the ratio achieved with NCV threshold 90 is still too big for a practical use, and the problem here is that a further increase of the NCV threshold would make the spam bulk detection even more vulnerable at the higher obfuscation ratios (see Figure 2(b)).

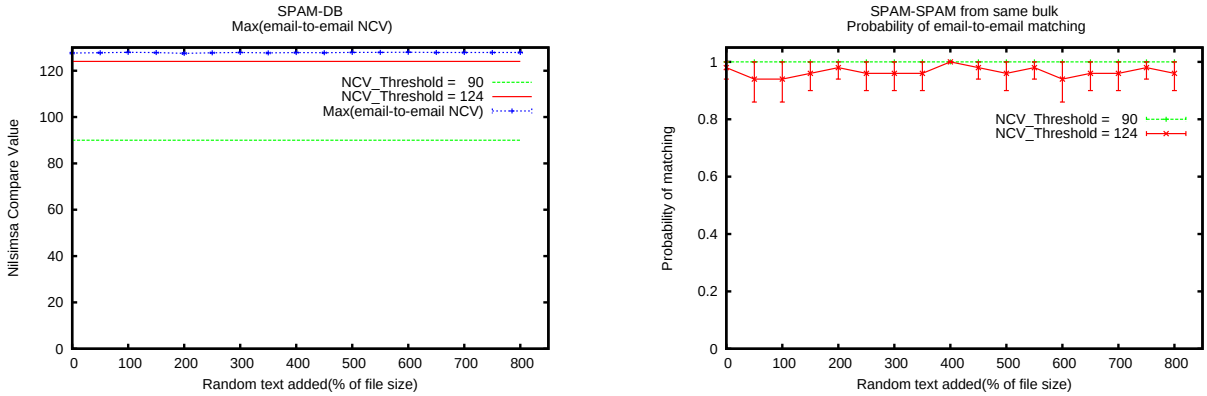
**Shifted and wide NCV histogram phenomena.** The NCV histograms between good emails and the emails from the DB database (Figure 5) show that the digests from non related emails are far from being bit-wise non-correlated (i.e. the bits set to 1 for a randomly chosen digest are not uniformly distributed over the digest positions), as the complete histograms are shifted to the right and not centered around the zero. This comes from the fact that the trigrams from the used language are not uniformly distributed.

The digests are also not independent, as the histogram is not completely gaussian and contains an additional local maximum. The dependence might come from the fact that there are many good emails that quote other emails (or their parts) in the replies, or simply there are unrelated good emails that discuss around the same currently popular topic. Quoting is especially exhibited a lot in discussions over mailing lists. The dependence normally makes the histogram wider than it would be for independent good emails. A lot of the right part on the Figures 5(a)-5(b) is populated, which implies use of high NCV threshold values to ensure low miss-detection of good emails, and makes a lot of the digests computed out of spam bulk emails (Figures 3(a)-3(d) ) ineffective for mutual matching, especially under a strong obfuscation of spam emails.

**What we can learn from the NCV histograms.** The NCV histograms and the above analysis show that the assumptions used in OD-paper, for an approximate estimation of the miss-detection probability by use of the Binomial probability distribution ($B(n,p)$, with n=256, and p=0.5), are far from being even approximately correct.

The above histogram analysis also gives some suggestions on how the conflict between the low miss-detection of good emails and good detection of spam bulk requirements could possibly be lessen. One suggestion, in a search for a way to ensure a smaller overlap between the HAM-DB NCV histograms and the SPAM-SPAM_BULK NCV histograms, is to try to use longer digests (e.g. of 512 bits, instead of 256 bits).

Another suggestion, that we actually evaluate in this paper, is to try to make the digest more specific by computing them not from complete emails (with or without headers), but from smaller email parts, e.g. from the strings of the constrained length sampled from the emails.




(a) (b)

**Figure 6: Alternative digest technique: Bulk spam detection under the "adding random text" spammer model. We can see from (b) that the spam bulk detection is very resiliant to the increased obfuscation effort by spammer. We can see from (a) that, similar as in the previous experiments, the average NCV is not a very informative metric.**

# 3. ALTERNATIVE TO THE ORIGINAL OPEN-DIGEST

As it is demonstrated in the previous experiments, the original open digest technique is vulnerable to a simple obfuscation by spammer. In this section we consider use of digests that are created from the strings of fixed length, sampled from an email at random positions. The constrained length of the samples from which the digests are produced should make it more difficult for spammers to easily "hide" spam text by adding a lot of random text into the email. Sampling strings at random positions should make the digests less predictable by the spammer, which should make aimed attacks (analyzed in OD-paper) less efficient. We experimentally compare the two ways of producing digests under the same conditions, i.e. assuming the same simple detection algorithm (the one described in Section 1.1.2).

OD-paper also considers use of multiple digests per email, but these are not created from strings of fixed length, and are not randomized. They are created the same way as in the single hash case, but only the different hash functions are used for each (the set of the used hash functions is assumed to be fixed and globally known).

## 3.1 Sampling strings and producing digests

The sampled strings are 60 characters in length, which looked to us as a good compromise between covering the spam phrases, and not giving to the spammer a lot of space for obfuscation. The initial string is sampled starting at a uniform random position between 1 and 30. For each new string the starting position is increased by 30 plus a random number between 1 and 30. We have chosen the parameters intuitively and didn't optimize them.

The digests are produced out of the sampled strings using the same Nilsimsa similarity hashing as in the single digest experiments.

## 3.2 Email-to-email comparison

We keep the same experiments as in the case with single digest, with the only difference in email to email comparison. We define, for the considered multiple digest approach, the NCV between two emails to be the maximum NCV over all the pairs of the digests between the two compared emails. The goal is to score how much similar are the most similar parts in the two emails.

## 3.3 "Spam bulk detection - new digest" experiment ($SPAM - DB$, *new digest*)

The experiment is the same as in the corresponding single digest case, i.e. as specified in Section 2.4, with the only difference in how the digests are produced and how the email to email comparison is performed (which is explained in Sections 3.1 and 3.2).

### 3.3.1 *Parameters and results discussion*

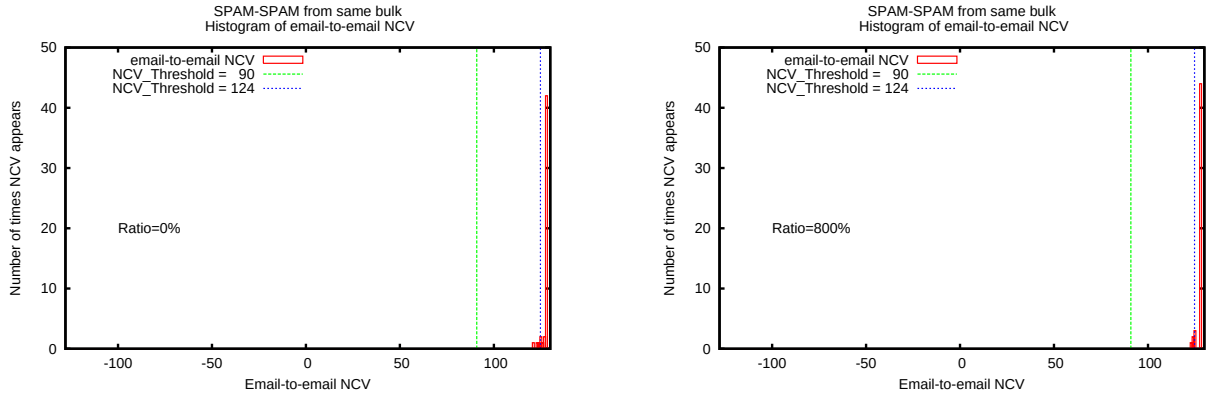
The results are shown in Figures 6 and 7.

**NCV threshold values.** We perform the experiments for the NCV threshold value 90, as it looked as a more reasonable choice between the two values we used in the single digest experiments. After observing the NCV histogram results, we find that there is a space for an additional increase of the NCV value. In order to illustrate the effect of the NCV change in the new digest experiments, we also perform them with the NCV threshold value 124. Again, we do not optimize the threshold.

**Spam detection is efficient and very resistant to the obfuscation.** As we can see from Figure 6(b), the spam detection is very efficient and resistant to the increased spam obfuscation. From Figure 7 we see that the digests useful for spam bulk detection became (as compared to the single digest case) very specific and practically not affected by the increased obfuscation (for the considered spammer model). As now the digests encode email parts of the constrained (actually fixed) length, they cannot be polluted a lot by the considered random characters addition attack (the behavior should be similar for the random readable (good) text addition attack).

## 3.4 "Good emails miss-detection - new digest" experiment ($HAM - DB$, *new digest*)

The experiment is the same as in the corresponding single digest case, i.e. as specified in Section 2.5, with the only difference in how the digests are produced and how the email to email comparison is performed (which is explained in Sections 3.1 and 3.2).




(a) (b)

**Figure 7: Alternative digest technique: Impact of increased obfuscation by spammer on mutual matching of emails from the same bulk.** **A random pair of emails from the same bulk match each other (with a high probability, as shown in Figure 6(b)) through the digests that are very specific to each other, and what is the most important the effective digests stay very specific even in the presence of very strong obfuscation under the considered spammer model. This allows for setting of the NCV threshold to high values, as needed for low miss-detection of good emails.**

### 3.4.1 *Parameters and results discussion*

The results are shown in Figures 8 and 9.

**Miss-detection of good emails kept similar as in OD-paper.** While the alternative digests provide bulk spam detection that is more efficient and much better resistant to the considered obfuscation, the obtained good email miss-detection results are similar as in the single digest case (NCV threshold values 124 and 90 in the alternative digests case correspond to the threshold values 90 and 54 in the single digest case, respectively). However the provided email-to-email (estimated) probabilities are rather too high for a practical use and needs to be further decreased.

**Opened a new dimension for further decrease of the miss-detection of good emails.** We made the alternative digests from the relatively short strings of fixed length in order to make them more specific and able to show whether two emails contain very similar parts (e.g. a message the spammer wants to get through within obfuscated emails) or not. This helped decreasing the overlap between the HAM-DB NCV histograms and the SPAM-SPAM_BULK NCV histograms.

When using the alternative digests, the fact that different digests encode different parts of an email allows to further lessen the mentioned histograms overlap, and so to lessen the conflict between the good detection of spam and low miss-detection of good email requirements. This should be possible to achieve e.g. by using so called "negative selection" AIS algorithm (see for example [4], [6]; AIS stands for Artificial Immune Systems) to first compare the digests computed from the newly received emails to a database of known good digests ("SELF" database), and eliminate those new digests that match, and only then compare the remaining digests from the new email (if any) with those in the database built through the collaborative bulk detection. In the terms of the NCV histograms, this would cause most of the right part or the HAM-DB NCV histogram to disappear. In practical terms, this should directly decrease the miss-detection of good emails.

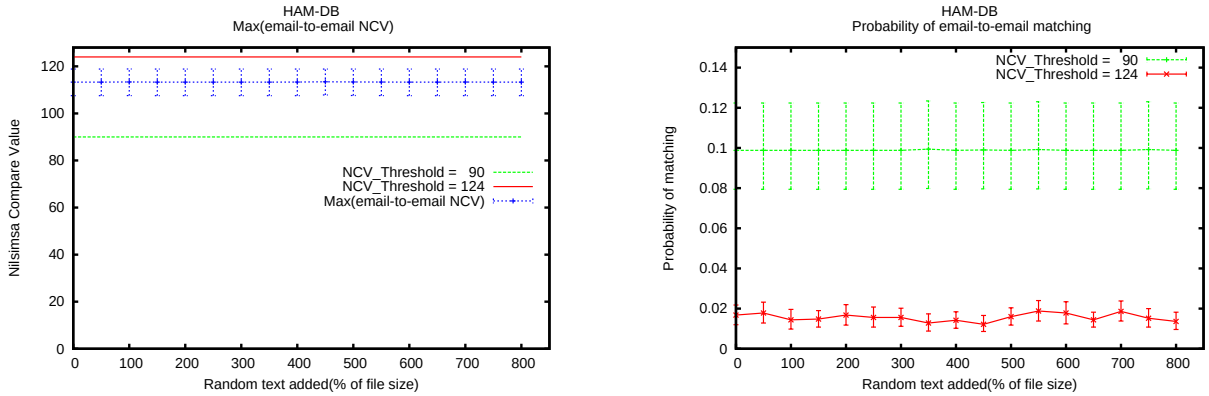
An important fact here is that in the single digest case, as the negative selection is applied on all new emails - so also on spam emails, the digests from spam emails that have a lot of good looking content (e.g. intentionally added by spammers) would also often be deleted, and the corresponding spam emails not detected (see "if any" in the previous paragraph). However, with the relatively short alternative digests deleting of the digests that include the "innocent" email patterns still leaves the high probability for survival of the digests that include the "novel" (not known as innocent) and possibly spammy patterns, which means preserving the detection of spam bulk at a good level. The possibility to keep some and eliminate other email parts opens a new dimension within the collaborative spam bulk detection.

Examples of innocent patterns that can be pre-collected or automatically extracted and updated to the "SELF" database are sender's email client information and used email formatting information, which are often present in email headers. Also, it is possible to build the ham content profile of a user and use it in the negative selection, when filtering emails for that user.

## 4. CONCLUSION

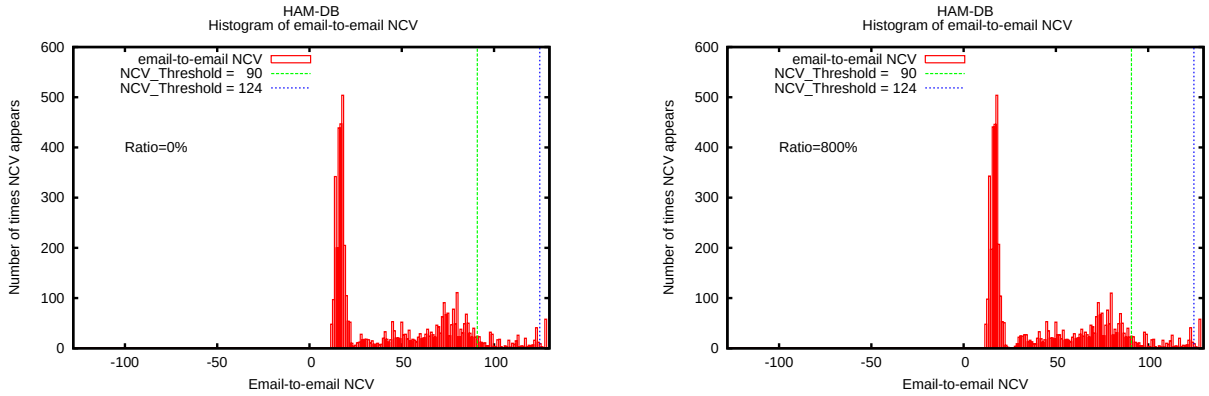
**Improved detection resistance to obfuscation.** We repeated and extend some of the open-digest paper [2] experiments, using the simplest spammer model from that paper. We found that the conclusions of the open-digest paper are rather miss-leading. Contrary to the findings of the OD-paper, the original digest technique actually is vulnerable to the increased obfuscation that is rather easy by the spammer to perform. We also found that the miss-detection of good emails is much higher then advocated (and partially experimentally supported) in OD-paper, for the NCV threshold that they propose and use (though a higher threshold is also not a good solution, as it would make the obfuscation of spam even easier). We proposed and evaluated, under the same spammer model, a modified version of the original digest technique. The modified version greatly improves the resistance of spam detection against increased obfuscation effort by spammers, while keeping miss-detection of good emails at a similar level.

**Alternative digests provide a good promise for further decreasing the miss-detection of good emails: negative selection.** Due to the property that different digests encode different parts of an email, the alternative digests look very promising regarding the possibility for inclu-




(a) (b)

**Figure 8: Alternative digest technique: Miss-detection of good emails under the "adding random text" spammer model.** **We consider the case in which the database DB of digests contains both spam and ham digests (the same case is considered in the OD-paper for the OD-paper digest). For the values of the NCV threshold that provide good spam bulk detection (which is resilient to the increased obfuscation - this was not possible to achieve with OD-paper digest), the miss-detection has very similar values to those obtained with the OD-paper digest technique.**




(a) (b)

**Figure 9: Alternative digest technique: Impact of the NCV threshold on the (undesirable) matching of good-email digests to the database of digest from both spam and good emails.** **We see that the NCV threshold should be set very high in order to provide low miss-detection of good emails, but with the alternative digests that is possible due to good detection of spam with high threshold values. We can also see that the obfuscation of spam has no effect on miss-detection of good emails.**

sion of additional algorithms into the basic detection scheme considered in this paper. E.g., so called negative selection algorithm could be added to first compare the digests computed from the newly received emails to a database of known good digests ("SELF" database), and eliminate those new digests that match, and only then compare the remaining digests from the new email (if any) with those in the database built through the collaborative bulk detection. This is aimed at decreasing the miss-detection of good emails. The NCV histogram results suggest that such a technique would be much more effective with the alternative digests then with the digests as in OD-paper (see Section 3.4 for a detailed explanation).

**Negative selection should be quantitatively evaluated** for its advocated ability to decrease the miss-detection of good emails.

**Other spammer models.** In this paper we did consider only one obfuscation (spammer model), the one to which the OD-technique looked very vulnerable, and we experimentally confirmed the vulnerability. While we show that the alternative digests are resistant to the considered obfuscation, they should be also tested under other obfuscation techniques, e.g. those considered in the OD-paper.

**Digest length.** As we already mentioned, it would be interesting to see whether use of longer digests (e.g. of 512 or 1024 bits, instead of 256) would help to decrease the overlap between the HAM-DB NCV histograms and the SPAM-SPAM_BULK NCV histograms, and so lessen the the conflict between the low miss-detection of good emails and good detection of spam bulk requirements.

**Similarity hashing variants.** With a goal to better counter the aimed addition attack (described in OD-paper, not evaluated here), it would be interesting to investigate the effect of replacing the use of the accumulators within the similarity hashing (used to produce the digests, and explained in detail in Section 1.1.2) by a bloom-filter like procedure for setting the bits of the digest. With this procedure, the bits to which the trigrams point are set to 1, regardless whether they were set previously or not; collisions are allowed but should not be too many.

## 5. REFERENCES


[1] Source code of the experiments from this paper: http://icawww.epfl.ch/ssarafij/improving-open-digest/, March 2008.

[2] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. An open digest-based technique for spam detection. In *Proceedings of The 2004 International Workshop on Security in Parallel and Distributed Systems*, San Francisco, CA, USA, September 2004.

[3] DCC. http://www.dcc-servers.net/dcc/, Feb 2008.

[4] J. Kim and P. J. Bentley. An evaluation of negative selection in an artificial immune system for network intrusion detection. In L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 1330–1337, San Francisco, California, USA, 7-11 2001. Morgan Kaufmann.

[5] Nilsimsa. http://lexx.shinn.net/cmeclax/nilsimsa.html, Sep 2006.

[6] S. Sarafijanovic and J.-Y. Le Boudec. Artificial immune system for collaborative spam filtering. In *Proceedings of NICSO 2007, The Second Workshop on Nature Inspired Cooperative Strategies for Optimization, Acireale, Italy, November 8-10, 2007.*

[7] SpamAssassin. http://spamassassin.org/, Feb 2008.

[8] SpamAssassin-Public-Corpus. http://spamassassin.org/publiccorpus/, Feb 2008.

[9] F. Zhou, L. Zhuang, B. Zhao, L. Huang, A. Joseph, and J. Kubiatowicz. Approximate object location and spam filtering on peer-to-peer systems. In *Proceedings of ACM/IFIP/Usenix Int'l Middleware Conf., LNCS 2672, pp. 1-20.*