

R&D Centre for Mobile Applications (RDC)  
FEE, Dept of Telecommunications Engineering  
Czech Technical University in Prague

**RDC Technical Report TR-13-3**

Internship supervisor: Lukas Kencl, lukas.kencl@fel.cvut.cz

# Evaluation of Information-Concealing Performance in Email Filtering

---

Chao Chhaya, Ecole des Mines d'Alès, France,  
chhaya.chao@mines-ales.org



Prague, September 2013

# Acknowledgments

I would like to express my special gratitude and thanks to Dr. Kencl who guided and supervised me during all my internship. He helped me in doing this project where i came to know about so many new things, i am really thankful to him.

I would like also to express my gratitude to my school department director Mr. Runtz, and to Prof. Bestak who gave me the golden opportunity to do my internship in Prague.

My thanks and appreciations also go to my colleague who have helped me out with their abilities, in developing the project.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	The Research and Development Centre . . . . .	1
1.2	The project: information-concealing in spam filtering . . . . .	2
1.3	Objectives . . . . .	2
<b>2</b>	<b>Related work</b>	<b>4</b>
2.1	Concealing information: algorithm . . . . .	4
2.2	Tools . . . . .	5
2.2.1	Postfix: server mail . . . . .	5
2.2.2	ThunderBird: client mail . . . . .	6
2.2.3	SpamAssassin: spam detection . . . . .	6
<b>3</b>	<b>Architecture</b>	<b>8</b>
3.1	Postfix-SpamAssassin associating: . . . . .	9
3.2	Postfix-ThunderBird associating: . . . . .	11
<b>4</b>	<b>Method to evaluate spams score evolution</b>	<b>12</b>
4.1	General Processes . . . . .	12
4.2	Solution . . . . .	14
4.2.1	Evaluation protocols . . . . .	14
4.2.2	Protocols automation . . . . .	15
<b>5</b>	<b>Perfomance evaluation</b>	<b>19</b>
5.1	SpamAssassin's score evolution with default SpamAssassin's ruleset . . . . .	19
5.2	SpamAssassin's score evolution with only dictionary ruleset . . . . .	22
5.3	Using of adapted dictionary ruleset . . . . .	24
<b>6</b>	<b>Conclusion</b>	<b>28</b>
	<b>Bibliography</b>	<b>29</b>
	<b>Appendices</b>	<b>30</b>

# List of Figures

3.1	Tools architecture . . . . .	8
3.2	Postfix and SpamAssassin functioning . . . . .	10
3.3	ThunderBird getting mail . . . . .	11
4.1	usual process to deliver mail and detect spam via SpamAssassin . . . . .	13
4.2	method process used to deliver mail and detect spam via SpamAssassin . . . . .	13
4.3	ThunderBird getting SpamAssassin scores . . . . .	14
5.1	Mail2 SA's score evolution with default ruleset . . . . .	20
5.2	Mail20 SA's score evolution with default ruleset . . . . .	20
5.3	Mail16 SA's score evolution with default ruleset . . . . .	20
5.4	Mail14, 17, 18 and 19 SA's score evolution with default ruleset . . . . .	21
5.5	Mail19 SA's score evolution with default ruleset and with only dictionary rules . . . . .	23
5.6	Mail14 SA's score evolution with default ruleset and with only dictionary rules . . . . .	23
5.7	Mail4 and mail17 SA's score comparison between unchanged dictionary rules (blue) and adapted dictionary rules for k=3,4,5 (red) . . . . .	25
5.8	Mail3 and mail15 SA's score comparison between unchanged dictionary rules (blue) and adapted dictionary rules for k=3,4,5 (red) . . . . .	26
5.9	Mail1 and mail20 SA's score comparison between unchanged dictionary rules (blue) and adapted dictionary rules for k=3,4,5 (red) . . . . .	27

## Résumé

Dans le domaine du réseau notamment, il arrive que les chercheurs soient amenés à partager leurs informations de réseaux. En effet certain algorithme de gestion de réseau requierent l'accès à ce genre d'information, spécialement dans le domaine de sécurité réseau et des systèmes de détection d'intrusion. Il s'agit d'un problème à cause de la possibilité de révéler des informations sensibles d'ordre privées ou professionnel.

Ce rapport traite de la méthode de dissimulation d'information développée par le laboratoire «Research and Development Centre» à Prague. Cette méthode consiste à comparer le «degré de spam» de mail par rapport à leurs différentes versions dissimulées. Celles ci se différencient avec un paramètre  $k$  qui refère à la longueur des blocs (en nombre de caractères) que l'on souhaite préserver à partir du mail originel.

Dans cette optique on utilise un server mail local Postfix et un module de détection de spam : SpamAssassin. Ce module est celui qui détermine le degré de spam des mails en associant un score a chaque mail. Plus ce score est important, plus le mail est considéré comme un spam. Il est alors possible de déterminer comment le score évolue avec la valeur de  $k$ .

L'objectif principal de ce projet est d'étudier la détection de spam à partir de leurs versions dissimulées.

## Abstract

In the domain of network in particular, it happens that researchers have to share some of their networking information. Indeed, many networking algorithm require access to this kind of information, especially in the domain of network security and intrusion detection systems. This is a problem due to the possibility of revealing sensitive information of private or business nature.

This project proposes a method to evaluate performance in mail filtering domain of the concealing information algorithm developed by the «Research and Development Centre» in Prague. This method consists on compare the «spam level» of mails to those of its different concealing version. These one differ with a  $k$  parameter which refers to the length of information we want to preserve from the original mail.

In this purpose we use a Postfix local server mail and a spam detection module: SpamAssassin. This module is the one which determines the spam level of mails with associated at each mail a score. Higher is this score, higher is the spam level. It is then possible to determine how the score evolves depending on the  $k$  value.

The main objective of this project is to study spam mail detection from concealed mail version.

# Chapter 1

## Introduction

For the second year at l'Ecole des Mines d'Alès, my internship takes place at the Research and Development Centre (RDC) laboratory in Prague, under the responsibility of my internship supervisor: Lukas Kencl, researcher and director of this RDC lab.

### 1.1 The Research and Development Centre

Research and Development Centre for Mobile Applications is a university laboratory based in the Czech Technical University (CTU) in Prague. This laboratory is focused in the domain of mobile networks and services and is in close collaboration with some industrial partner like Vodafone and IBM.

The main mission of this laboratory is to deliver internationally competitive research results in services and technologies in the area of mobile wireless networking, with results of high value to industrial partners.

So all the projects developed in this laboratory are about mobile networks, and we can in particular name the actual projects deal by the RDC lab [9]:

- Network Technology, Mobility and Security: Protection against attacks in IP telephony.
- Voice Services: Extraction of information from Web in order to present it to a user using speech recognition and synthesis.
- 3D Mobile Internet: The project focuses on various topics of mobile computer graphics and virtual reality.
- e-Scribe: Design and set up an online voice transcription centre for the hearing-impaired.
- Cloud Computing: There are project about cloud data security and cloud latency.

The concealing information project is associated with all the projects which are about sharing information. That's why it has a direct interest in the domain of cloud computing where people are able to share sensitive information and where it could be interesting to conceal this one.

## **1.2 The project: information-concealing in spam filtering**

It is a real problem for network researchers in particular to store and make available, for studies, their information like networking traces containing entire packets payloads etc... Indeed, this is difficult due to possibility to share sensitive information, while protection of the sensitive content is crucial for extensive information sharing.

During this project we will study a new method designing an anonymizing technique to conceal information [4]. The advantage of this method is to make impossible to reconstruct the initial information from the anonymized output content, but still enable some search into it (malicious keyword for example).

In fact the concept of hiding information potentially sensitive has been studied recently in various Information and Communication Technologies subdomains. There is for example the steganography method which refers to the art and science of writing hidden messages in such a way that no one apart the sender and the intended recipient know that there is a hidden message.

This method has direct application in various domain like in cloud computing where the actual problem is that people have to trust company which store data. Another application concerns the domain of mail where we want a structure detecting spams without having direct access to mail itself which contains potential sensitive information.

## **1.3 Objectives**

This internship has multiple objectives. First, it is about to understand the stakes of the project and its benefits. Then, in a technical point of view, objectives are to familiarise with the concealing information algorithm and its Matlab code associated. Afterward it will be question of set up a protocol to evaluate performance of the algorithm and automatize this protocol thanks to scripts for example.

In fact the main objective of this internship is to verified some properties announced by the concealing information method or the necessary parameters to make these properties true. We can in particular name the property which refers to preservation of local information (malicious keyword of certain maximal length for example) and the potential

preservation in size (initial information and anonymized output content could have similar size in bits).

For this study we especially work on the preservation of some locals information. In this purpose we study this property using mail support to know if a mail detected like a spam could also be detected as, after being concealed by the method.



# Chapter 2

## Related work

### 2.1 Concealing information: algorithm

The input of the algorithm is a sequence. This one could be a text or an information sequence from a wave, video etc...

First of all it is necessary to turn this sequence into a cycle: the end of the sequence is connected to its beginning. Then, the concealing itself is based on repeats. Indeed, the algorithm is constituted of 5 procedures:  $S, S^1, S^{1+}, S^2$  and  $S^{2+}$ .

The basic idea of all these procedures is quite the same [4]. There is the input cyclic sequence  $\omega$  which is partitioned into consecutive disjoint blocks. Then, in front of each block we add the terminal part of the preceding block. This adding part is called the *overlap*. The resulting sequence contains all the studied local information. Depending on the procedure, these segments would contain some excess information that are vital in composition of the procedures and that is the key in this concealing algorithm. Furthermore an additional segment can be added (but not needed) behind each block: it is the *dust*.

The enhanced blocks are called the *cards* and the last step consist on shuffling these cards to obtain the output cyclic sequence  $\omega_F$ .

Here an example of the S procedure which takes these following parameters:  $\omega$  the input sequence,  $o$  the overlap length,  $lb$  the lower bound of the length of a block and  $ub$  the upper bound of the length of a block. Consider the input sequence as «the aim of this paper is to present an information-concealing method ». To make this example easier to understand, let's replace the empty-space by «E». Finally, we have these parameters: Input  $\omega = \mathbf{t h e E a i m E o f E t h i s E p a p e r E i s E t o E p r e s e n t E a n E i n f o r m a t i o n E c o n c e a l i n g E m e t h o d}$ ,  $o=3$ ,  $lb=4$  and  $ub=6$ .

$S(\omega, o=3, lb=4, ub=6)$  procedure example:

First the input sequence  $\omega$  is partitioned into disjoint blocks of size include between 4 ( $lb$  parameter) and 6 ( $ub$ ). To make it easy to read we separate the blocks by a «+»:  
**theEai+mEofEt+hisE+pape+rEisE+toEpr+esent+EanEi+nfor+matio+nEco+ncea+lingE+algor+ithm+**

Then the overlap of length 3 is added in front of each block:  
**thmtheEai+EaimEofEt+fEthisE+isEpape+aperEisE+isEtoEpr+Epresent+entEanEi+nEinfor+formatio+tionEco+Econcea+cealingE+ngEalgor+gorithm+**

Next the dust is added behind each block (of length approximately 2):  
**thmtheEaip+EaimEofEtim+fEthisEcon+isEpapeEin+aperEisEa+isEtoEproEp+Epresentese+entEanEilgo+nEinforiE+formatiofo+tionEcoE+EconceaEci+cealingEpa+ngEalgorEp+gorithmap+**

Finally cards are shuffling to be arrange in a random order. Then this following output is obtained:

**ngEalgorEpnEinforiEformatiofocealingEpaaperEisEafEthisEconEconceaE  
cithmtheEaipisEtoEproEpisEpapeEinEpresentesetionEcoEentEanEilgogor  
ithmapEaimEofEtim**

## 2.2 Tools

### 2.2.1 Postfix: server mail

Postfix is a free and open source mail transfer agent, developed by Wietse Venema in 1997. The original main objective of this software was to propose an alternative to the Sendmail software. The main Postfix features are to be fast, easy to administer, secure, while being as far as possible compatible with Sendmail. In fact, Postfix objectives can be listed [3]:

- Large diffusion: Postfix has to be largely diffuse and use, to have a real impact about performance and security of messaging system on the Internet. That's why it is a free and open source software, with no restriction.
- Performance: Postfix is at least three times faster than his bigger rival «Qmail». On a computer, a Postfix server can daily send/receive one million of different mails.
- Security: Postfix uses several security levels to protect system from intrusion.
- Safety and robustness: When system has no more memory or free space, Postfix will not cause damage, it was developed to be under control.

- Compatibility: Postfix was developed to be compatible with Sendmail to facilitate the software switch.
- Flexibility: In fact Postfix contains several programs with their associated tasks. Each of this program could be replacing by an other one developed by users.

## 2.2.2 ThunderBird: client mail

Mozilla ThunderBird is a client mail free and open-source developed and distributed by the Mozilla Foundation.

It is used to read and send mails. Here is the list of some of its features [6]:

- Message management: Thunderbird can manage multiple email, newsgroup and news feed accounts and supports multiple identities within accounts.
- Extension and themes: ThunderBird allows the addition of features via add-ons.
- Security: Thunderbird provides some default security features and others can be added through extensions.
- Filtering: Thunderbird incorporates a Bayesian spam filter, a whitelist based on the included address book etc...
- Multiple platform support: Thunderbird runs on a wide variety of platforms (Windows, Linus, OS X, OpenSolaris etc...).

## 2.2.3 SpamAssassin: spam detection

Spamassassin is also a free and open-source software released by the Apache Software Foundation [11].

The aim goal of Spamassassin is to filter mails to detect spams. It is compatible with a lot of server mail like Procmal, Sendmail, Postfix, Qmail etc... and it can be installed on most of system based on Windows, Linux and Mac OS.

SpamAssassin applies a large set of rules to determine if a mail is a spam or not. In fact, according to results of these test, Spamassassin attributes a score to the mail and if this one is higher than the required score that is defined in a Spamassassin configuration file, then the mail is detected like a spam.

Under Linux, this Spamassassin configuration file is located at `/etc/spamassassin/local.cf` and it looks like this:

```
rewrite_header Subject [***** SPAM _SCORE_ *****]
required_score          2.0
#to be able to use _SCORE_ we need report_safe set to 0
#If this option is set to 0, incoming spam is only modified by adding some
#"X-Spam-" headers and no changes will be made to the body.
report_safe            0
# Enable the Bayes system
use_bayes              1
use_bayes_rules        1
# Enable Bayes auto-learning
bayes_auto_learn       1
# Enable or disable network checks
skip_rbl_checks        0
use_razor2             0
use_dcc                0
use_pyzor              0
```

On the first two lines we can define the required score to consider a mail as a spam (here it is 2 for example) and the manner how we change header of mail detected as spam. Then, the «report\_safe 0 »means that spamassassin will just change header of spam without changing content of body. Finally, all the other parameters like «use\_bayes », «use\_bayes\_rules »etc... set at 1 or 0, define activation or not of the associated parameters.

We assume that for this project, we will only be interested to the required score part, and let all the other parameters by defaults.

# Chapter 3

## Architecture

This section is about how Postfix Spamassassin and ThunderBird work together. For our project, here is the different ways emprunted by a mail:

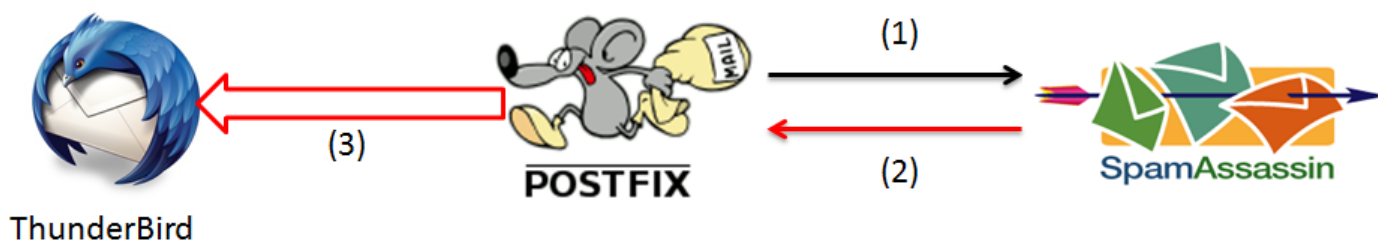


Figure 3.1: Tools architecture

(1)First, when a mail is delivered on the Postfix server mail, this one is transmitted to SpamAssassin.

(2)Then, SpamAssassin treats this mail and return it to Postfix with its associated score depending on rules that are activated.

(3)Finally, ThunderBird gets the mail and its score which always appears in the header of the mail in our case.

When Postfix, Spamassassin and ThunderBird have been well installed, this implementation is made in two steps: the Postfix-SpamAssassin associating and the Postfix-ThunderBird associating.

### 3.1 Postfix-SpamAssassin associating:

After the installation of Postfix and SpamAssassin, we want to make them work together. In this purpose we have first to create a new group and user for SpamAssassin. Indeed, by default SpamAssassin runs as its own user, which is not optimal. Then, we create the group and user «spamd» with its home directory, as root user:

```
groupadd -g 5001 spamd
useradd -u 5001 -g spamd -s /sbin/nologin -d /var/lib/spamassassin spamd
mkdir /var/lib/spamassassin
chown spamd:spamd /var/lib/spamassassin
```

Next it has to associate SpamAssassin to the user «spamd». This step is done by modifying this following lines in the file `/etc/default/spamassassin`:

```
ENABLED=1
OPTIONS="--create-prefs --max-children 5 --helper-home-dir"
PIDFILE="/var/run/spamd.pid"
```

to:

```
ENABLED=1
SAHOME="/var/lib/spamassassin/"

OPTIONS="--create-prefs --max-children 5 --username spamd --helper-home-dir
${SAHOME} -s ${SAHOME}spamd.log"

PIDFILE="${SAHOME}spamd.pid"
```

What happens here, is that we are going to run SpamAssassin as user `spamd` and make it use its own home dir (`/var/lib/spamassassin/`) and is going to output its logs in `/var/lib/spamassassin/spamd.log`. Moreover, «`ENABLED=1`» means that `spamassassin` daemon is allowed to start.

Afterward it is question to make Postfix using SpamAssassin by modifying the file `/etc/postfix/master.cf`, where it is necessary to replace:

```
smtp      inet  n       -       -       -       -       smtpd

by:

smtp      inet  n       -       -       -       -       smtpd
        -o content_filter=spamassassin
```

And of course, in the same file, we have to define «spamassassin», depending on its user previously associated. So at the end of the file let's add:

```
spamassassin unix -      n      n      -      -      pipe
      user=spamd argv=/usr/bin/spamc -f -e
      /usr/sbin/sendmail -oi -f ${sender} ${recipient}
```

Finally, we have to start SpamAssassin and Postfix to make all this settings available:

```
/etc/init.d/spamassassin start
/etc/init.d/postfix reload
```

The above command will actually reload Postfix and start `spamd`, a daemonized version of SpamAssassin, which is much quicker than the official Perl version as it actually loads all SpamAssassin rules once at startup.

We assume that SpamAssassin has been already configured (required score etc...). If not it is necessary to configure the required score before to start or restart SpamAssassin.

**Note:** Because we always want the score even if the mail is not a spam, this one is fixed at -1 in `/etc/spamassassin/local.cf`:

```
required_score      -1
```

At the end of all this steps Postfix runs according to this schema [1]:

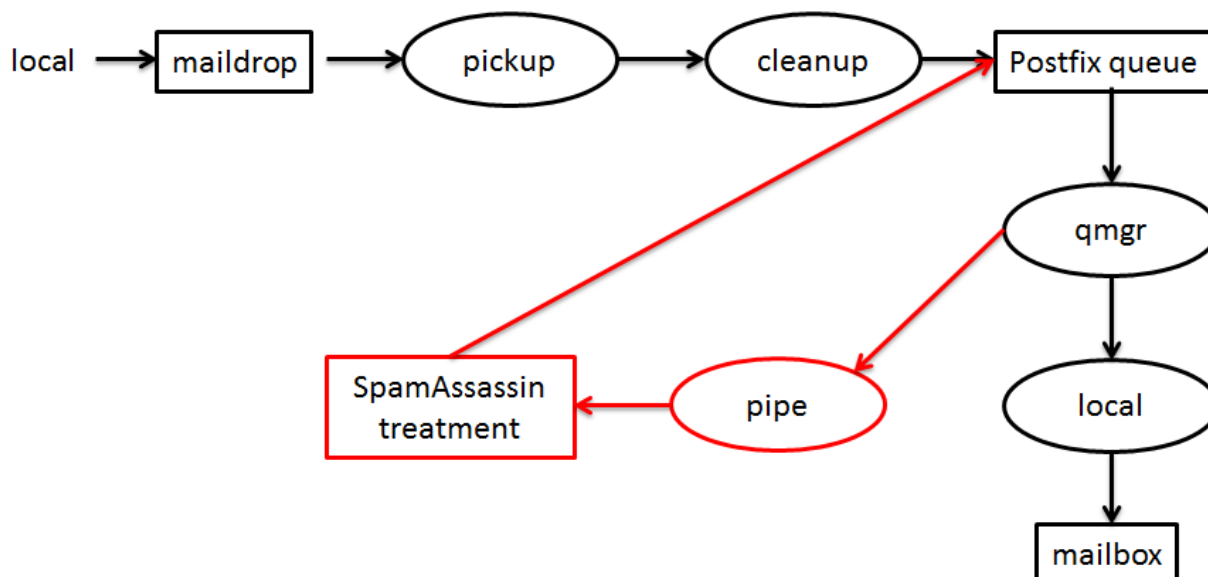


Figure 3.2: Postfix and SpamAssassin functioning

Without SpamAssassin, when a local mail arrives in the maildrop directory, the Postfix programs «pick up»and «cleanup»post it in the Postfix queue. Next «qmgr»which periodically scan the queue deal the mail to the program «local»which deliver the mail in mailbox.

Now associating SpamAssassin to Postfix, the mail is interfaced with SpamAssassin -thanks to the «pipe»program- which analyse and attribute a score to the mail before to post it again in the Postfix queue. When the mail is analysed and in the Postfix queue, it can be deliver in mailbox.

### 3.2 Postfix-ThunderBird associating:

In this part it is question to associate Postfix to ThunderBird.

First it is necessary to add the account on ThunderBird which propose 3 different kinds of account: mail account, chat account and other account. So we choose to add a «Unix Mailspool»in the «other account»section. Finally, when the account is well added ThunderBird can get all mails which arrive in the mailbox.

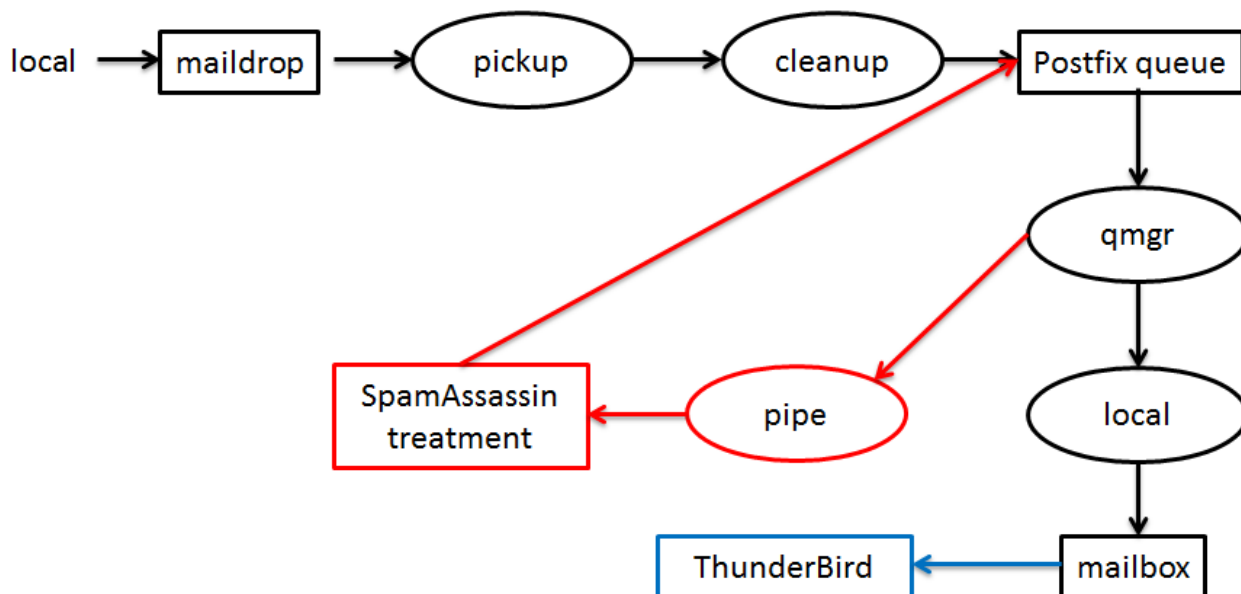


Figure 3.3: ThunderBird getting mail



# Chapter 4

## Method to evaluate spams score evolution

The purpose of this chapter is to propose a way to analyze the concealing method performance. Let us remind that this study is established using the mail support (cf: 1.3, Objectives).

Before beginning with more technical details, it is important to notice that the solution has to work under the Linux environment. This is the only restriction concerning the solution setup. Indeed the choice of softwares and the manner to realize the solution under Linux are open.

### 4.1 General Processes

According to the part 2.2, some software have been chosen: Postfix, Mozilla Thunder-Bird and Spamassassin.

The idea is first to use Postfix to create a local server mail on our machine to have possibility to send and receive mails from the Linux terminal. Then, coupling Spamassassin to Postfix, each mail is associated with a score. So it is possible to get on the one hand the Spamassassin score of an initial mail and on the other hand the score of its concealed version associated. So it is question to compare this score, and determine if the initial mail and its concealed version have similar property relative to Spamassassin.

Of course it is interesting to concealed the initial mail with different parameters (k values especially) to determine parameters for an optimal local information preservation.

On a general manner here is how we will deal with mails:



Figure 4.1: usual process to deliver mail and detect spam via SpamAssassin

On this usual case the mail to deliver is treated by SpamAssassin and deliver either to normal mailbox or to spams mailbox, depending on results of SpamAssassin analysis. For this project, a concealing step is inserted before the SpamAssassin treatment:



Figure 4.2: method process used to deliver mail and detect spam via SpamAssassin

Because it is more convenient, mail and its associated SpamAssassin score is always deliver to the normal mailbox.

Finally, associated Postfix to the client mail ThunderBird is a way to facilitate sending and reception of mails. Indeed, Postfix is not practical to use from a Linux terminal because of all commands to enter to send a mail. However, with ThunderBird we can work with our local server very easily thanks to the simple graphical interface of this software.

## 4.2 Solution

### 4.2.1 Evaluation protocols

The aim of this protocol is to get score of an initial mail and score of its conceal version associated. For that purpose we first have to conceal the mail thanks to the matlab code developed by the RDC laboratory. Different version of this concealing can be executed with different parameters, for example  $k=3,4,5\dots$

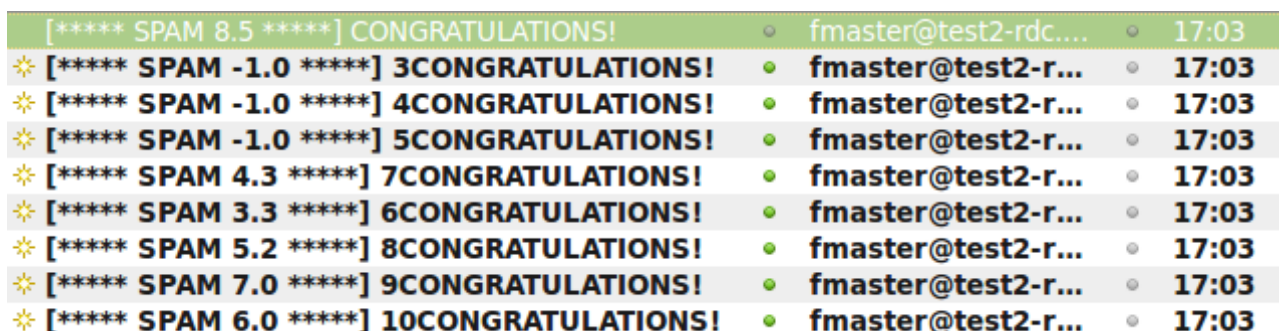
Here an example of typical command to conceal a file using the Matlab code developed by RDC:

```
p='/home/chhaya/Bureau/scrip_mail/IC'  
addpath('$p'),  
IC_install('$p')  
inputFormat='txt'  
inputFile='../input_ex_mails/ex_mail1'  
outputFile='../output_ex_mails/1/concealed_k3_ex_mail1'  
k=3  
options=IC_options(k,'weak')  
state=IC_state('random')  
N=inf
```

```
output=IC_concealFile(inputFormat,inputFile,outputFile,options,state,N)
```

The main function is «IC\_concealFile»but it is necessary to precise all the parameters used by this one: the path of the work directory, the name of input file (file to conceal) and output file (name of concealed file), type of input etc... and the k value which refers to the length of sequences we want to preserve from original mail.

Then the idea is to send the initial mail and its different concealed version thanks to thunderbird and so get the score of each of them.



[***** SPAM 8.5 *****] CONGRATULATIONS!	fmaster@test2-rdc...	17:03
☀ [***** SPAM -1.0 *****] 3CONGRATULATIONS!	fmaster@test2-r...	17:03
☀ [***** SPAM -1.0 *****] 4CONGRATULATIONS!	fmaster@test2-r...	17:03
☀ [***** SPAM -1.0 *****] 5CONGRATULATIONS!	fmaster@test2-r...	17:03
☀ [***** SPAM 4.3 *****] 7CONGRATULATIONS!	fmaster@test2-r...	17:03
☀ [***** SPAM 3.3 *****] 6CONGRATULATIONS!	fmaster@test2-r...	17:03
☀ [***** SPAM 5.2 *****] 8CONGRATULATIONS!	fmaster@test2-r...	17:03
☀ [***** SPAM 7.0 *****] 9CONGRATULATIONS!	fmaster@test2-r...	17:03
☀ [***** SPAM 6.0 *****] 10CONGRATULATIONS!	fmaster@test2-r...	17:03

Figure 4.3: ThunderBird getting SpamAssassin scores

This different steps are a good way to have first results. However, what it is interesting here it is to have a certain quantity of results which allow some statistical analyze. That is why it is necessary to automatize this protocol in the purpose to have a lot of results with a very fast and practical method.

## 4.2.2 Protocols automation

This subsection exposes the way to automatize protocols show in the last part. In this purpose bash script have been used.

All the commands could be containing in the same script, but to make it easier to understand we choose to split script into two scripts: the first one is to automatically conceal some mails located in a specified repertory and the second one is to send the mails and its different concealing-version associated.

**-Script to conceal mails:** This script is the one that involves all the matlab part. Like all bash script, it begins with the line which precise that it is a bash script:

```
#!/bin/bash
```

Then, a loop will be used to conceal a certain number of mails. For example, if we want to conceal 20 mails, the loop begins at 1 and finishes at 20:

```
num_mails=20
for ((i = 1; i <= $num_mails; i += 1))
do
```

Afterwards all the parameters have to be specify in bash variables (cf part 3.1.3). These parameters are those that setup the Matlab code and those that are used in the function to conceal a file:

```
p='/home/chhaya/Bureau/scrip_mail/IC'
inputFormat='txt'
inputFile=./input_ex_mails/ex_mail$i
outputFile=./output_ex_mails/$i/concealed_k3_ex_mail$i
k=3
N=inf
```

Finally, matlab is involved thanks to a bash command, and the function to conceal a file is executed. Furthermore the loop ended:

```
/usr/local/matlab/bin/matlab -nosplash -nodesktop -nojvm -r "addpath('$p'),
IC_install('$p'), options=IC_options($k,'weak'), state=IC_state('random'),
output=IC_concealFile('$inputFormat','$inputFile','$outputFile',options,state,$N),
quit"
```

```
done
```

Note: The «-nosplash»and «-nodesktop»means that Matlab will not be running graphically. The «-nojvm»option is used to reduce the memory requires because we know that the java-virtual-machine features are not necessary in our script. And the «-r»means that we want to run a file into Matlab.

Of course it is possible to execute other concealing with other parameters. For instance to conceal also the mail in a k=5 version, it is necessary to add this following command before to end the loop:

```
outputFile=../output_ex_mails/$i/concealed_k5_ex_mail$i
k=5

/usr/local/matlab/bin/matlab -nosplash -nodesktop -nojvm -r "addpath('$p'),
IC_install('$p'), options=IC_options($k,'weak'), state=IC_state('random'),
output=IC_concealFile('$inputFormat','$inputFile','$outputFile',options,state,$N),
quit"

done
```

However, for the project it would be interesting to conceal a mail with a k value included between 3 and 10 (or more). In this purpose it is enough to add a second loop to manage the k value evolution and modify a bit the outputFile variable:

```
num_mails=20
for ((i = 1; i <= $num_mails; i += 1))
do

#second loop to manage k evolution:
for ((k = 3; k <= 10; k += 1))
do

p='/home/chhaya/Bureau/scrip_mail/IC'
inputFormat='txt'
inputFile=../input_ex_mails/ex_mail$i
# k value inserting in the outputFile name
outputFile=../output_ex_mails/$i/concealed_ $k\$_ex_mail$i
N=inf

etc...
```

Finally this final script select all the 20 files containing in the «input\_ex\_mails»directory (ex\_mail1, ex\_mail2... ex\_mail20), and conceal each of this mail in a specific repertory in «output\_ex\_mails»with k values included between 3 and 10.

**-Script to send mails:** This script is used to send mails with Postfix, thanks to bash command. As the first script, it begins with the line that precises that it is a bash script, and defines some variables:

```
#!/bin/bash

srv_ip=127.0.0.1
srv_port=25
recipient=chhaya@localhost
```

Then, a first loop manages the sending of the 20 initial mails:

```
num_Mails=20
for ((i = 1; i <= $num_Mails; i += 1))
do

#the content of the mail is the content of the file ex_mail1 etc..
my_message='cat input_ex_mails/ex_mail$i'
subject='head -n 1 input_ex_mails/ex_mail$i'
nc $mail_srv_ip $mail_srv_port << EOF
ehlo mail.script
mail from:<fmaster@test2-rdc.org>
rcpt to:<$recipient>
data
Subject: $subject
$my_message
.
quit
EOF
```

The two first lines of this part define the body and the header of the mail: the body is the text containing in the file ex\_mail1, ex\_mail2,..., ex\_mail20 and the header is the first line of these text files. It assumes that on the first line of each of these 20 initial mails, there is the subject title of the current mail. Then, the mail server IP, mail server port, the sender and the recipient are in particular specified.

The «data»means that the writing of the mail itself will follow and this step is done thanks to the variables «subject»and «my\_message». Finally, the dot marks the end of the writing and then we quit Postfix.

Before to end the loop, it is necessary to add a second loop to send the concealed version of each ex\_mail1, ex\_mail2,... and ex\_mail20:

```
for ((k = 3; k <= 10; k += 1))
do

my_message='cat output_ex_mails/concealed_${k}\$_ex_mail$i'
subject='head -n 1 input_ex_mails/ex_mail$i'
nc $mail_srv_ip $mail_srv_port << EOF
[...etc...]
EOF
#end of the second loop
done

#end of the first loop
done
```

Note: Only the body of the mail changes because the first line of concealed text file will not represent the header and it is better to keep a representative header's mail.

To summarize this script sends one by one an initial mail (ex\_mail1, ex\_mail2,... and ex\_mail20). When one of this mail is sent, it sends also the concealed version of this one with a k value included between 3 and 10.

It is then possible to get all of these mail on thunderBird to read their scores. In the case where mails not arrive in the order they were sending, it is necessary to find a way to recognize which mail is associated at which value of k. In this purpose it is for example possible to change a bit the header of each sending message.

Considering the initial mail (non conceal) subject as unchanged it is enough to change the header of all the concealed mails:

```
subject='head -n 1 input_ex_mails/ex_mail$i'
```

is changed to:

```
subj='head -n 1 input_ex_mails/ex_mail$i'
#concatenation
subject=${k}$subj
```

# Chapter 5

## Performance evaluation

### 5.1 SpamAssassin's score evolution with defaults ruleset

The first results were done without changing the SpamAssassin's ruleset.

The following results had been obtained by concealing 20 spams with a value of k including between 3 and 10:

Concealing version	Original	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
Mail 1	16.1	0.2	0.8	2	7.6	7.6	12.6	15.1	12.6
Mail 2	30.9	0.2	0.2	3.5	10.9	17.6	23.2	24.8	25
Mail 3	11.6	0.3	7.5	7.7	2.8	9.5	9.5	11.6	12.3
Mail 4	35.7	3	3	7.1	11.2	12.2	16.2	7	13.7
Mail 5	2.7	0.2	5	6	3.3	2.8	2.7	2.7	2.7
Mail 6	26.2	1.8	5.1	9.1	5.7	17.6	20.7	19.7	21.7
Mail 7	9.4	0.2	0.2	0.2	0.2	3.5	1.6	0.2	3.5
Mail 8	4.2	0.2	0.2	0.2	4.3	1.9	4.2	4.3	4.3
Mail 9	31.2	5.1	1.8	8.7	9.8	13.3	20.9	10.5	25.2
Mail 10	3.4	1.3	0.2	1.3	4.1	0.2	6.9	8	10.7
Mail 11	17.4	3	6.3	6.5	11.3	17.4	13.8	12.9	16.3
Mail 12	33.5	2.8	2.8	8.6	2.8	13.7	15.2	12.7	15.2
Mail 13	6	0.2	0.2	0.2	0.2	1.1	5.1	5.1	2.6
Mail 14	15	1.8	2.4	6.5	2	7.5	12.5	12.5	12.5
Mail 15	20.7	3	3.6	4.8	10.2	8.7	11.1	18.2	14.6
Mail 16	15	1.8	1.8	2.4	8.5	12.5	10.9	12.5	12.5
Mail 17	15.1	0.2	0.2	7	7	10.1	14.1	11.6	15.1
Mail 18	11	1.8	3.2	7.5	7.5	10	10.3	14.8	11.5
Mail 19	15.1	2.8	4.6	7.4	9.5	12.6	15.1	12.6	12.6
Mail 20	19.8	2.8	2.8	4.5	5.9	11.6	15.2	15.2	15.2

Table 5.1: SpamAssassin's score evolution with k using default SpamAssassin's ruleset



From this results it appears three kinds of behavior. First, there are some mails - as mail 2 an 20 - which have the behaviour expected: the SpamAssassin score increases continually or stagnates with the value of k. Indeed, this should be logical because bigger is the value of k better is the conservation of words, and so, better is the detection of forbidden spam word for SpamAssassin.

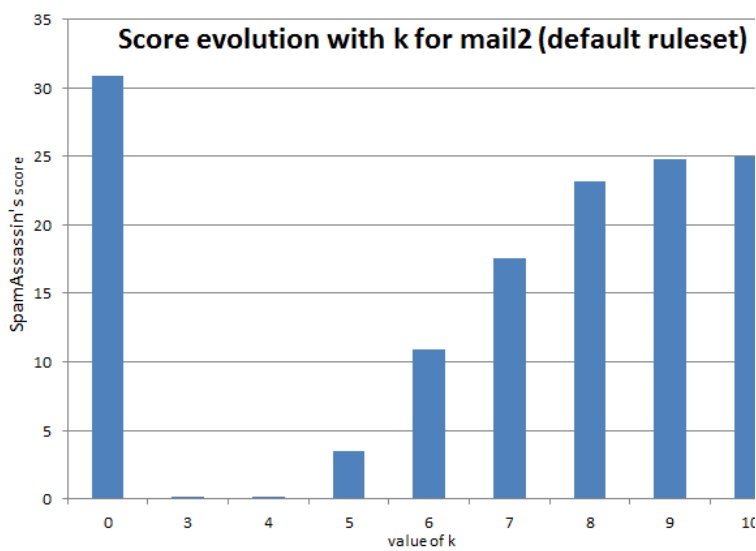


Figure 5.1: Mail2 SA's score evolution with default ruleset

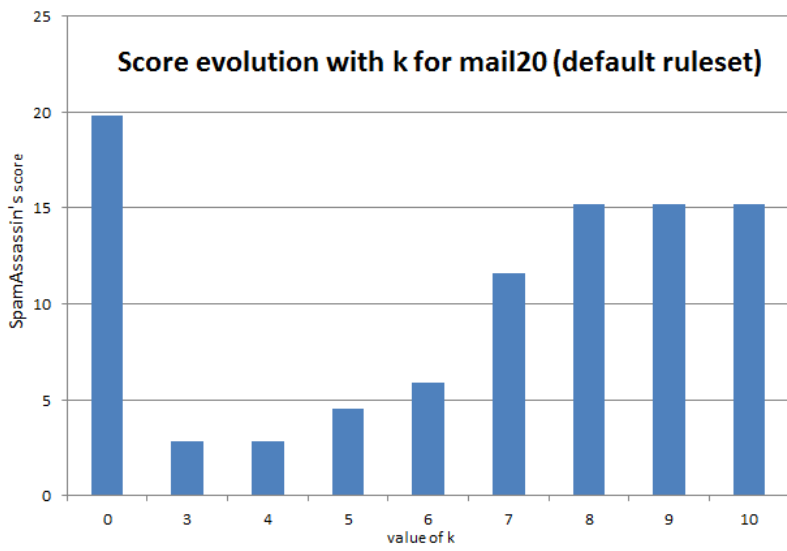


Figure 5.2: Mail20 SA's score evolution with default ruleset

Note: The k=0 refers to the score of original mails (without concealing).

Then, there is a second kind of behavior where the score decreases a bit at a local value of k, compare to the score at k-1:

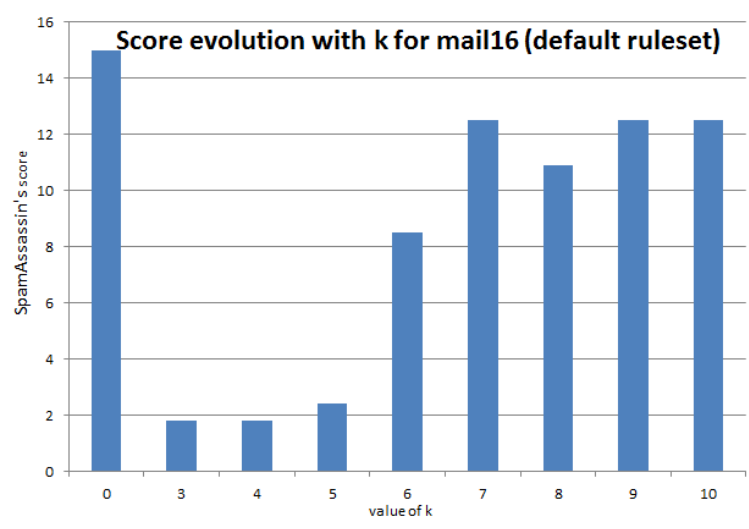


Figure 5.3: Mail16 SA's score evolution with default ruleset

This kind of decrease is «acceptable» because it does not significantly change scores. This decrease depends on single rule which will not be hit at local value of  $k$ . Indeed, depending on how cards were shuffled, and so, how they were rearranged, it is possible to hit a rule at a  $k$  value and not to hit at  $k+1$ .

However, there are also some mails which have an unexpected behavior: cases where decrease of score is too big to just involve only one rule (ex: mail14, mail17), or the decrease does not appear for a local value of  $k$  (ex: mail19), or the score at a  $k$  value is higher than the score of original mail (ex: mail18).

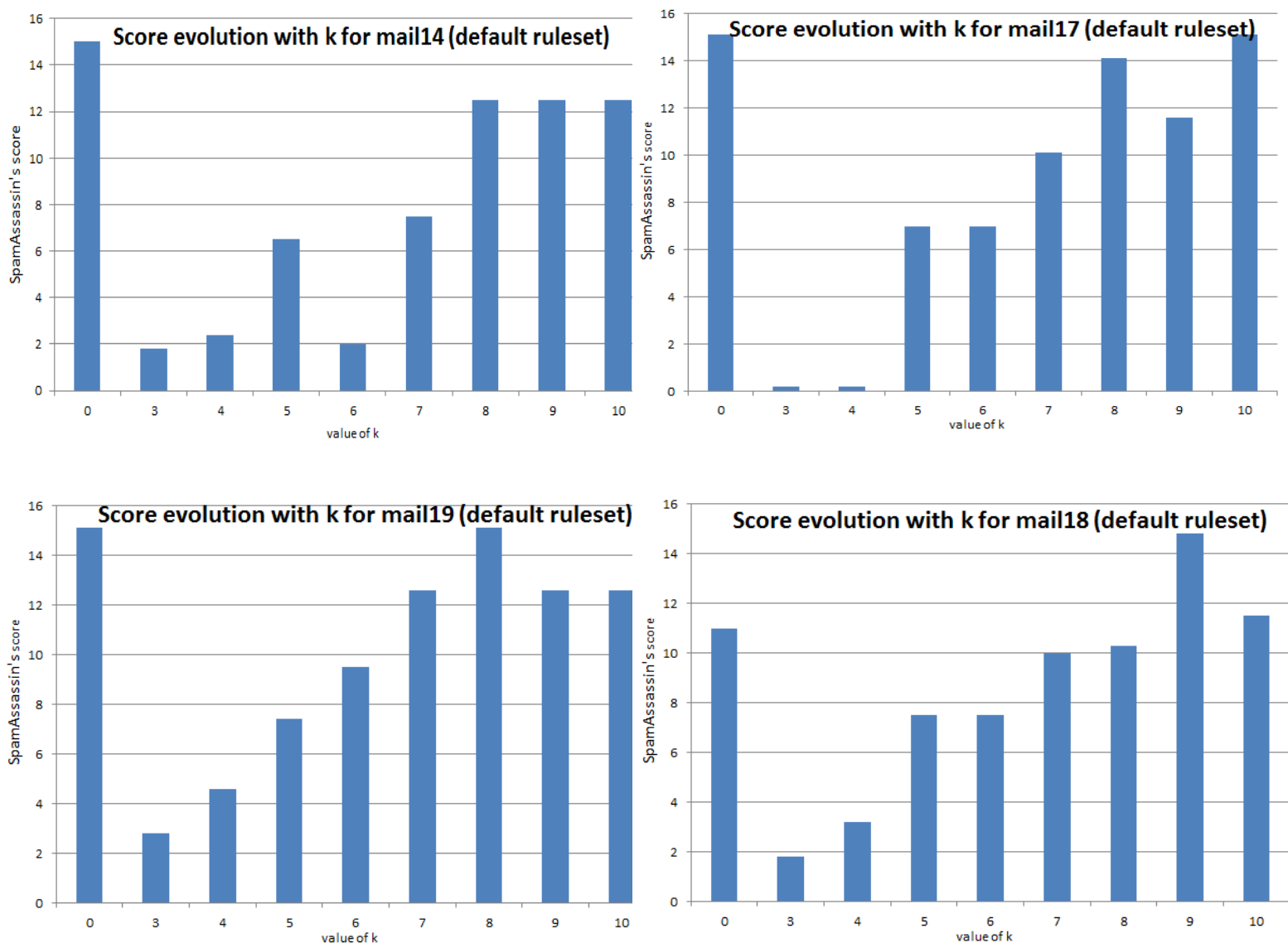


Figure 5.4: Mail14, 17, 18 and 19 SA's score evolution with default ruleset

This third behavior in particular appears because SpamAssassin does not only use dictionary rules containing forbidden words. Indeed, there are also some rules which

analyse form and structure of mails, some others are case-sensitive etc...

In this project we want to especially focus on information SpamAssassin contains in words, that is why it is more interesting to only work with dictionary SpamAssassin rules.

## 5.2 Score evolution with only dictionary ruleset

In this part it is a question to run SpamAssassin with only dictionary ruleset. In this purpose it is necessary to deal with some SpamAssassin configuration files: `/usr/share/spamassassin/50_score.cf` and `/usr/share/spamassassin/72_score.cf`. These files define the score associated to each SpamAssassin rule. So, the idea is to set at 0 all the rules which are not dictionary like.

Most of the rules are located at `/usr/share/spamassassin`. It is where we have to check to know if a rule is dictionary like or not.

Because of the huge number of rules we do not allow all the dictionary rules in SpamAssassin. That's why with this new ruleset some of used mails get very low score and are not treated in this part.

Concealing version	Original	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10	k=20
Mail 1	5,7	-1	-1	-1	0,7	0,7	3,3	3,3	3,3	5,7
Mail 2	8,4	-1	-1	-1	3,3	4,3	5,2	6	6	8,4
Mail 3	9,3	-1	2,9	2,9	2,9	7,2	7,2	9,3	8,7	9,3
Mail 4	6,6	1,6	1,6	4,2	1,6	4,2	4,2	4,2	4,2	4,2
Mail 5	1,5	-1	1,5	1,5	1,5	1,5	1,5	1,5	1,5	1,5
Mail 6	7,1	0,6	0,6	1,1	0,6	1,1	2,1	2,1	2,1	4,6
Mail 9	8,8	0,6	0,6	0,6	6,3	3,1	3,7	3,7	6,3	8,8
Mail 11	4,3	1,6	1,6	3,4	4,3	4,3	4,3	3,4	4,3	4,3
Mail 12	7,8	1,6	1,6	3,3	1,6	5,8	5,8	5,8	5,8	7,8
Mail 14	5,6	0,6	0,6	0,6	0,6	0,6	3,1	3,1	3,1	5,6
Mail 15	7,6	1,6	1,6	1,6	4,1	1,6	1,6	5,1	5,1	7,6
Mail 16	5,6	0,6	0,6	0,6	3,1	3,1	3,1	3,1	3,1	5,6
Mail 17	5,7	-1	-1	4	4	3,3	5,7	5,7	5,7	5,7
Mail 19	5,8	1,6	1,6	1,6	3,3	3,3	3,3	3,3	3,3	5,8
Mail 20	5,8	1,6	1,6	3,3	3,3	3,3	5,8	5,8	5,8	5,8

Table 5.2: SpamAssassin's score evolution with k using dictionary ruleset

The first observation is that score are lower in that case, which is logical because only dictionary rules have been allowed. Furthermore, it appears that for these mails, the SpamAssassin score is continually increasing/stagnating - or have some acceptable decreases - until to reach the original score. Indeed, we observe that at  $k=20$ , we generally obtained the original score.

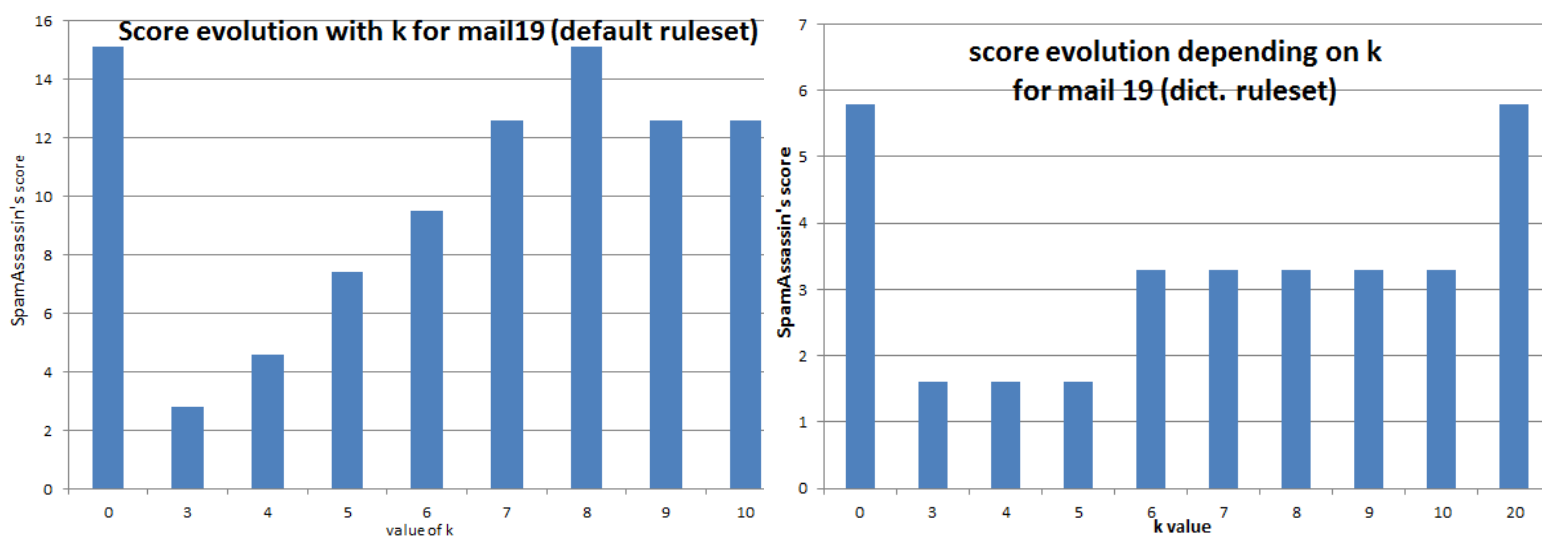


Figure 5.5: Mail19 SA's score evolution with default ruleset and with only dictionary rules

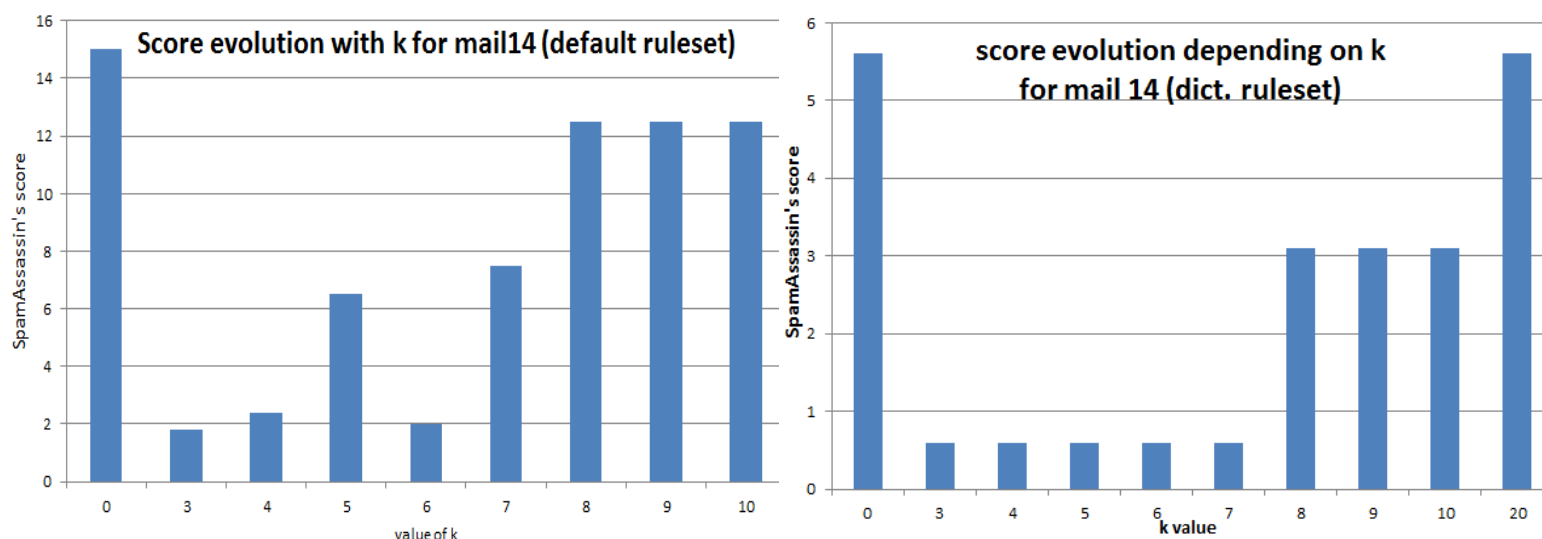


Figure 5.6: Mail14 SA's score evolution with default ruleset and with only dictionary rules

It emerges that with this new ruleset using only SpamAssassin dictionary-like rules, text analysis make possible to detect spams from a certain value of  $k$ . For instance in our case we can approximately set this value a  $k=8$ , where the score is not equal to the original one, but high enough to detect spams. Of course this value is to determine with a better method, with statistical study for example.

However, we would like to study if it is possible to make this analysis for lower value of  $k$ , at  $k=3,4$  for example. Indeed, lower is the value of  $k$ , better is the concealing.

Here we observe that for lower value of  $k$ , the SpamAssassin score is very low compared to the score of original mail. This observation is understandable because of how dictionary are written in SpamAssassin. These rules are not adapted to the concealing with low values of  $k$ .

### 5.3 Using of adapted dictionary ruleset

In this section the idea is to change the dictionary rules used in the previous part into rules adapted for low value of  $k$ . In this purpose we use the property of local information preservation of the algorithm. For example, if a rule detects the word «winner»we can adapt it for  $k=3$  with detecting all of these following sequences: «win», «inn», «nne»and «ner». And for  $k=4$ : «winn», «inne»and «nner».

So, after adapting these rules for  $k=3$ , then  $k=4$  and finally  $k=5$  (cf: appendix), we obtain these following results:

	unchanged dictionary	dict. adapted to $k=3$		dict. adapted to $k=4$		dict. adapted to $k=5$	
	Original	Original	$k=3$	Original	$k=4$	Original	$k=5$
Mail 1	5.7	6.6	7.6	6.6	6.6	6.6	6.6
Mail 2	8.4	9.9	9.9	8.4	8.4	8.4	8.4
Mail 3	9.3	9.3	13.9	9.3	9.3	9.3	9.3
Mail 4	6.6	6.6	6.6	6.6	6.6	6.6	6.6
Mail 5	1.5	5.6	13.9	3.9	3.9	1.5	1.5
Mail 6	7.1	7.1	7.1	7.1	7.1	7.1	7.1
Mail 9	8.8	8.8	11	8.8	8.8	8.8	8.8
Mail 11	4.2	6.7	11	6.7	6.7	6.7	6.7
Mail 12	5.7	5.7	5.7	5.7	5.7	5.7	5.7
Mail 14	5.6	5.6	5.6	5.6	5.6	5.6	5.6
Mail 15	7.5	9.2	9.2	7.5	7.5	7.5	7.5
Mail 16	5.6	5.6	6.6	5.6	5.6	5.6	5.6
Mail 17	5.7	5.7	5.7	5.7	5.7	5.7	5.7
Mail 19	5.7	6.2	11.4	5.7	5.7	5.7	5.7
Mail 20	5.7	8.2	11.8	8.2	8.2	8.2	8.2

Table 5.3: SpamAssassin’s score evolution with  $k$  using dictionary ruleset

What is important here is the score from the original mail using unchanged dictionary ruleset compare to score of concealed mails ( $k=3,4,5$ ) with their adapted dictionaries rulesets.

From these results we observe that for some mails the original score got with using unchanged dictionary ruleset is directly reach at  $k=3,4$  and  $5$  (mails 4, 6, 12, 14 and 17).

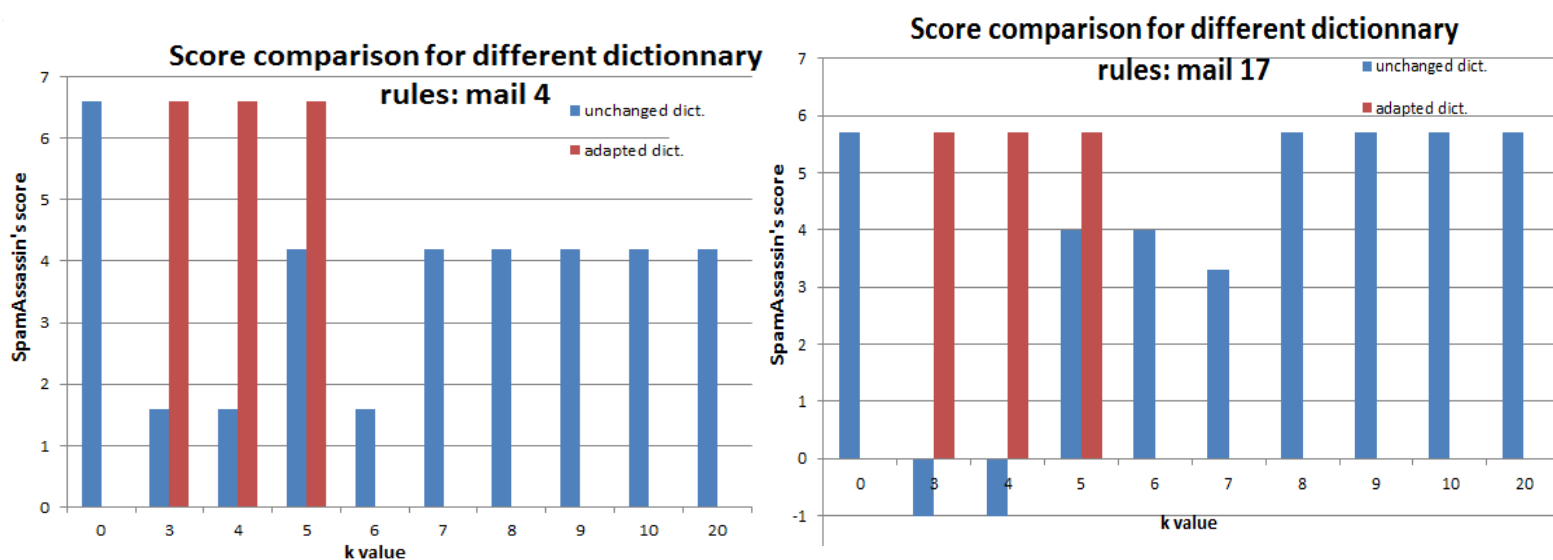


Figure 5.7: Mail4 and mail17 SA's score comparison between unchanged dictionary rules (blue) and adapted dictionary rules for  $k=3,4,5$  (red)

Then there is a second case:

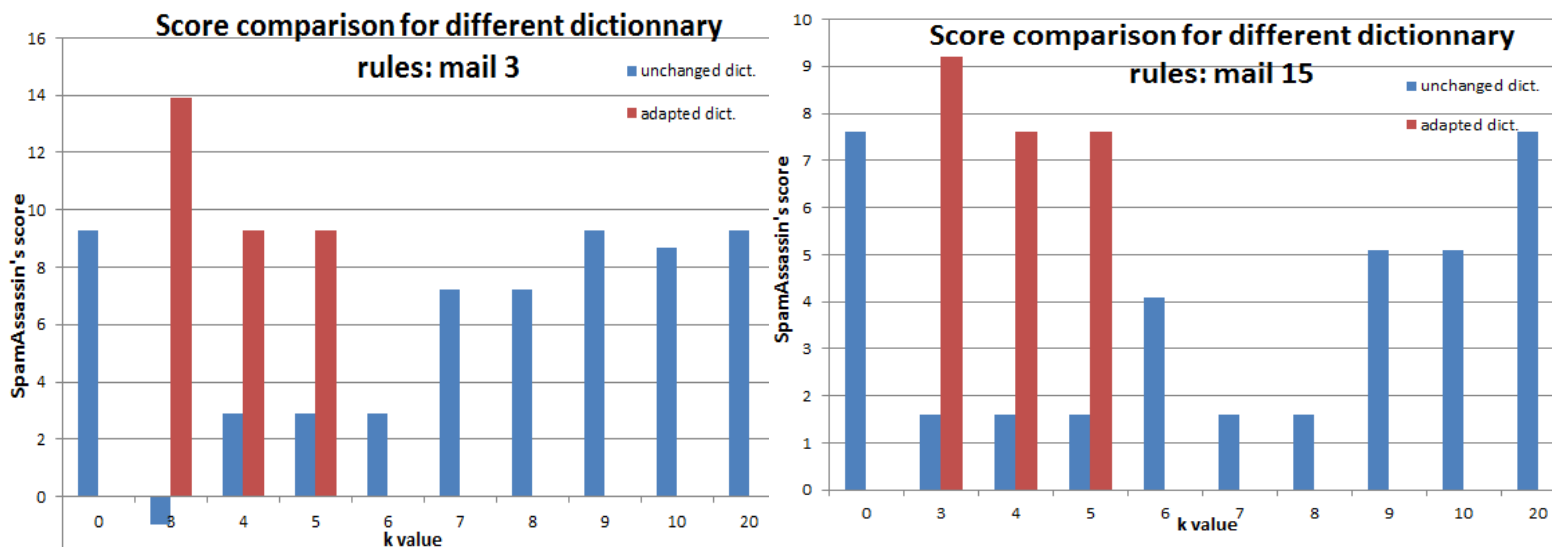


Figure 5.8: Mail3 and mail15 SA's score comparison between unchanged dictionary rules (blue) and adapted dictionary rules for k=3,4,5 (red)

Here the score of the original mail (with unchanged dictionary rules) and those with k=4,5 (with their adapted dictionary ruleset associated) are the same, but not for k=3 which has a higher score (cf mails 1, 2, 3, 9, 15, 16 and 19). This is due to what we call «false negative»: new words not present in an original mail appear in concealing version. For example in concealing version with k=3, because of the rearrangement after the card shuffling step, it is possible to find the sequences «win», «inn», «nne» and «ner» which hit the rule «winner», while the word «winner» is not present in the original mail. In that case the original mail do not hit the rule winner among the unchanged dictionary ruleset, while the concealing version with k=3 hit the associated and adapted rule.

Finally there is a last case where even dictionaries adapted at k=4 and k=5 are higher than the score get from original mail and unchanged dictionary ruleset:

This happens because of how some rules are adapted. Some of original dictionary-like rules detect associated words in sentence and not just single word separately. For example, a rule could be hit with detecting the words «millions» and «dollars» separated with less than 20 characters. However, because of the shuffling step - in particular - in the algorithm and because of k values which are used, we have to use simpler rules: the «millions dollars» rule is hit when it detects «millions» and «dollars» separately, adapted to k=3,4 and 5 of course (for example with k=3 it is sufficient to detect mil, ill lli etc... and dol, oll, lla etc...).

This part shows that score given by SpamAssassin -using adapted dictionaries rulesets- for mail concealing with low value of k are quite similar or equal to the score of the original mail (given by using only the unchanged dictionary rules).

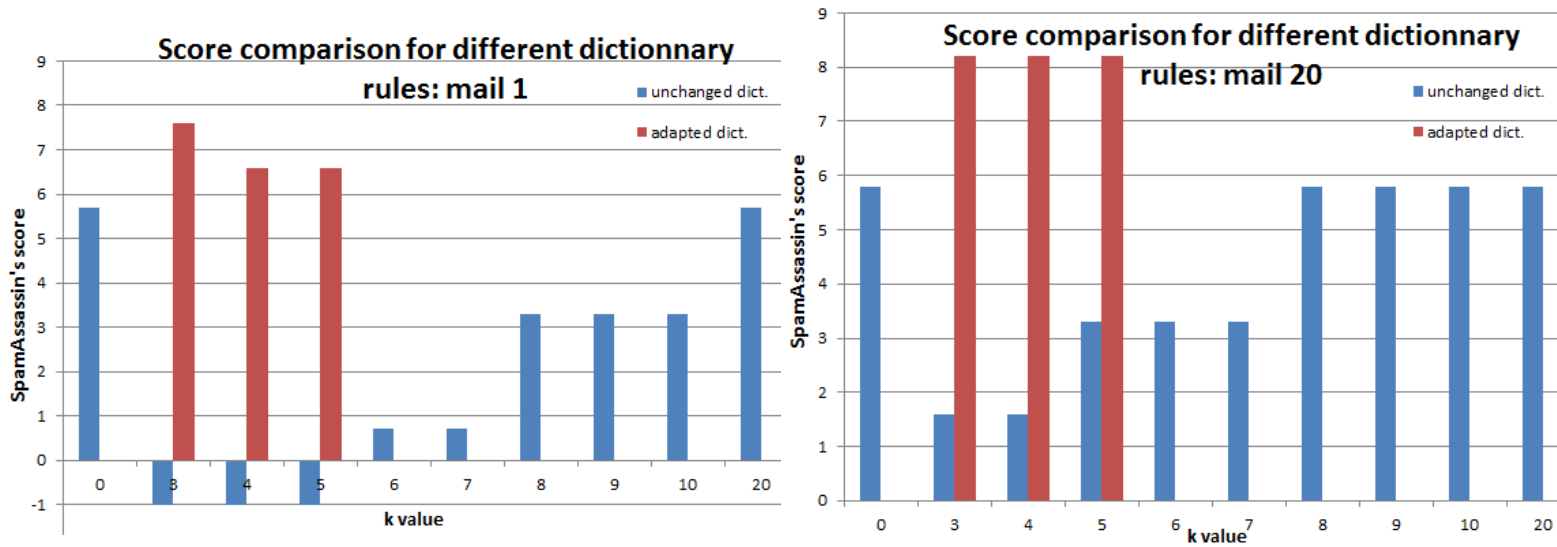


Figure 5.9: Mail1 and mail20 SA's score comparison between unchanged dictionary rules (blue) and adapted dictionary rules for k=3,4,5 (red)



# Chapter 6

## Conclusion

From the results of this project we show first that the concealing information algorithm - developed by RDC - preserves well local information. Indeed, in the default ruleset part we can observe that the SpamAssassin score generally increase with the k value in concealing text. However, for some certain case results expose an important decrease of the score for local value of k. This happened because of existence of rules which analyse in particular form and structure of mails.

This observation is confirmed when we only used dictionary ruleset. In that case we get better results because for most of our used mails, the score always increase or stagnate or present very low decrease, until to reach the score of original mail.

However, when only dictionary ruleset is been use, score are too low for low value of k. But, the fact is that lower the value of k is, better is the concealing. That's why it is necessary to use adapted dictionary ruleset.

Finally when adapted dictionary ruleset are used, fine results ermerge. Indeed, in that case it is possible to directly reach the score of original mail for  $k=3$  or 4.

All of this results tend to show that using the algorithm developed by RDC, it is possible to detect spams by analysing concealed version of these ones, even if concealing are made with short information length preservation.

In the future of this project it could be interesting to develop a complete and optimize dictionary ruleset in the purpose to confirmed our results using better environment parameters.

# Bibliography

- [1] AKADIA Information Technology AG. *Fighting Spam with SpamAssassin and Postfix*. [http://www.akadia.com/services/postfix\\_spamassassin.html](http://www.akadia.com/services/postfix_spamassassin.html).
- [2] Chantra. *Postfix and Spamassassin: How to filter spam*. <http://www.debuntu.org/postfix-and-spamassassin-how-to-filter-spam/>, 2006.
- [3] Cyril Jovet. *Postfix: la documentation*. <http://cjovet.free.fr/cours/postfix.htm>, 2002.
- [4] Lukas Kencl and Martin Loeb. *DNA-inspired information concealing: a survey*. *Elsevier Computer Science Review*, (4), jul 2010.
- [5] Lukas Kencl, Martin Loeb, and Jenny Blamey. *Processing of data information in a system*. *US Patent Application*, (US 2010/0205676 A1), aug 2010.
- [6] Mozilla. *ThunderBird features*. <http://www.mozilla.org/en-US/thunderbird/features/>.
- [7] José Zamora Ponce, Martin Loeb, and Lukas Kencl. *Packet content anonymization by hiding words*. In *Infocom 2006*, 2006.
- [8] Postfix. *The Postfix Home Page*. <http://www.postfix.org/>.
- [9] Research and Development Centre. *Projects*. <http://www.rdc.cz/en/projects/>.
- [10] Scriptdemo. *Run matlab m-file in a shell script*. <http://scriptdemo.blogspot.cz/2010/11/run-matlab-m-file-in-shell-script.html>, 2010.
- [11] SpamAssassin. *The Apache SpamAssassin Project*. <http://spamassassin.apache.org/>.
- [12] Ubuntu-documentation. *PostfixBasicSetupHowto*. <https://help.ubuntu.com/community/PostfixBasicSetupHowto/>.

# APPENDICES

## Appendix A: Script to send mails

```
----- Script to send mails -----
#!/bin/bash

#variables :
mail_srv_ip=127.0.0.1
mail_srv_port=25
recipient=chhaya@test2-rdc.org

#we will send « num_Mails » mails
num_Mails=20
for ((i = 1; i <= $num_Mails; i += 1))
do

#***** sending of original mails
#the content of the mail is the content of the file ex_mail1, ex_mail2, ... or ex_mail_10
my_message=`cat input_ex_mails/ex_mail$i`

#the subject is the first line of the file
subject=`head -n 1 input_ex_mails/ex_mail$i`

#commands lines to send mail :
nc $mail_srv_ip $mail_srv_port << EOF
ehlo mail.script
mail from:<fmaster@test2-rdc.org>
rcpt to:<$recipient>
data
Subject: $subject
$my_message
.
quit
EOF

#***** sending of concealed version with k included between 3 and 10
for ((k = 3; k <= 10; k += 1))
do
#Msg is the content of concealed3,...,concealed10 located in directory associated to original mail
my_message=`cat output_ex_mails/$i/concealed$k`

#the subject still is the first line of the file containing original mail
subj=`head -n 1 input_ex_mails/ex_mail$i`
#trick to rapidly recognize the version of concealed mail : 3subject, 4subject, ..., 10subject
subject=$k$subj

nc $mail_srv_ip $mail_srv_port << EOF
ehlo mail.script
mail from:<fmaster@test2-rdc.org>
rcpt to:<$recipient>
data
Subject: $subject
$my_message
.
quit
EOF

done
done
```

## Appendix B: Script to conceal mails

```
#!/bin/bash

#num_mails : number of file (ex_mail1, ex_mail2, ..., ex_mail20)
num_mails=20
for ((i = 1; i <= $num_mails ; i += 1))
do

#we conceal for k included between 3 an 10
for ((k = 3; k <= 10; k += 1))
do

#parameters :
p="/home/chhaya/Bureau/scrip_mail/IC"
inputFormat='txt'
inputFile=./input_ex_mails/ex_mail$i
outputFile=./output_ex_mails/$i/concealed$k
#N=inf means we conceal all text in the file
N=inf

# concealing step :
/usr/local/matlab/bin/matlab -nosplash -nodesktop -nojvm -r "addpath('$p'), IC_install('$p'),
options=IC_options($k,'weak'), state=IC_state('random'),
output=IC_concealFile('$inputFormat','$inputFile','$outputFile',options,state,$N), quit"

done
done
```

## Appendix C: Dictionary ruleset used

(Note that all of these rules are located in /usr/share/spamassassin/20\_phrases.cf or /usr/share/spamassassin/72\_active.cf)

body DEAR\_SOMETHING                    ^bDear (?:ITW|Internet|candidate|sirs?|madam|investor|travell?er|car  
shopper|web)\b/i  
describe DEAR\_SOMETHING                Contains 'Dear (something)'

body DEAR\_FRIEND                        /^\s\*Dear Friend\b/i  
describe DEAR\_FRIEND                    Dear Friend? That's not very dear!

body URG\_BIZ                             /urgent.{0,16}  
(?:assistance|business|buy|confidential|notice|proposal|reply|request|response)/i  
describe URG\_BIZ                         Contains urgent matter

body UNCLAIMED\_MONEY                    ^bunclaimed\s(?:assets?|accounts?|mon(?:ey|ies)|balance|funds?|prizes?|rewards?|payments?|deposits?)\b/i  
describe UNCLAIMED\_MONEY                People just leave money laying around

body US\_DOLLARS\_3                        /(?:\\$|usd).?d{1,3}[.,.]\d{3}[.,.]\d{3}(?:[.,.]\d\d)?/i  
describe US\_DOLLARS\_3                    Mentions millions of \$ (\$NN,NNN,NNN.NN)

body MILLION\_USD                         /Million\b.{0,40}\b(?:United States? Dollars?|USD)/i  
describe MILLION\_USD                     Talks about millions of dollars

body LOTTO\_AGENT                        ^b(?:claim(?:sing)?  
(?:\sprocessing)?|fiducia|w+|reimbursement(?:prize|international|intl|foreign|win+ing)(?:[s.,]+  
(?:rem+it+ance|settlement|payment|award|transfer))+|payment|immunity|grants?)s?  
(?:agent|manager|officer|secretary|director|mgr\b)/i  
describe LOTTO\_AGENT                    Claims Agent

body DEAR\_WINNER                        ^bdear.{1,20}winner/i

body IMPOTENCE                          ^b(?:impotence (?:problem|cure|solution)|Premature Ejaculation|erectile dysfunction)/i  
describe IMPOTENCE                        Impotence cure

body BODY\_ENHANCEMENT                    ^b(?:enlarge|increase|grow|lengthen|larger\b|bigger\b|longer\b|thicker\b|binches\b).  
{0,50}\b(?:penis|male organ|pee[ -]?pee|dick|sc?hlong|wh?anger|breast(?:!s+cancer))/i  
describe BODY\_ENHANCEMENT                Information on growing body parts

body BODY\_ENHANCEMENT2                   ^b(?:penis|male organ|pee[ -]?pee|dick|sc?hlong|wh?anger|breast(?:!s+cancer)).  
{0,50}\b(?:enlarge|increase|grow|lengthen|larger\b|bigger\b|longer\b|thicker\b|binches\b|size)/i  
describe BODY\_ENHANCEMENT2                Information on getting larger body parts

body BANG\_GUAR                          ^bguaranteed?!/i  
describe BANG\_GUAR                        Something is emphatically guaranteed

body GUARANTEED\_100\_PERCENT             /100% GUARANTEED/i  
describe GUARANTEED\_100\_PERCENT         One hundred percent guaranteed

## Appendix D: Dictionary ruleset adapted to k=3

body \_\_DEAR\_SOMETHING1 /Dea/i  
body \_\_DEAR\_SOMETHING2 /ear/i  
body \_\_DEAR\_SOMETHING3 /sir/i  
body \_\_DEAR\_SOMETHING4 /ar /i  
body \_\_DEAR\_SOMETHING5 /r s/i  
body \_\_DEAR\_SOMETHING6 / si/i

meta DEAR\_SOMETHING\_C (\_\_DEAR\_SOMETHING1 && \_\_DEAR\_SOMETHING2 && \_\_DEAR\_SOMETHING3 && \_\_DEAR\_SOMETHING4 && \_\_DEAR\_SOMETHING5 && \_\_DEAR\_SOMETHING6)  
score DEAR\_SOMETHING\_C 1.7  
describe DEAR\_SOMETHING\_C Contains 'Dear (something)'

body \_\_DEAR\_FRIEND1 /fri/i  
body \_\_DEAR\_FRIEND2 /rie/i  
body \_\_DEAR\_FRIEND3 /ien/i  
body \_\_DEAR\_FRIEND4 /end/i  
body \_\_DEAR\_FRIEND5 /r f/i  
body \_\_DEAR\_FRIEND6 / fr/i

meta DEAR\_FRIEND\_C (\_\_DEAR\_SOMETHING1 && \_\_DEAR\_SOMETHING2 && \_\_DEAR\_FRIEND1 && \_\_DEAR\_FRIEND2 && \_\_DEAR\_FRIEND3 && \_\_DEAR\_FRIEND4 && \_\_DEAR\_SOMETHING4 && \_\_DEAR\_FRIEND5 && \_\_DEAR\_FRIEND6)  
score DEAR\_FRIEND\_C 2.6  
describe DEAR\_FRIEND\_C Contains 'Dear friend'

body \_\_URG\_BIZ1 /urg/i  
body \_\_URG\_BIZ2 /rge/i  
body \_\_URG\_BIZ3 /gen/i  
body \_\_URG\_BIZ4 /ent/i  
body \_\_URG\_BIZ5 /bus/i  
body \_\_URG\_BIZ6 /usi/i  
body \_\_URG\_BIZ7 /sin/i  
body \_\_URG\_BIZ8 /ine/i  
body \_\_URG\_BIZ9 /nes/i  
body \_\_URG\_BIZ10 /ess/i

meta URG\_BIZ\_C (\_\_URG\_BIZ1 && \_\_URG\_BIZ2 && \_\_URG\_BIZ3 && \_\_URG\_BIZ4 && \_\_URG\_BIZ5 && \_\_URG\_BIZ6 && \_\_URG\_BIZ7 && \_\_URG\_BIZ8 && \_\_URG\_BIZ9 && \_\_URG\_BIZ10)  
score URG\_BIZ\_C 0.9  
describe URG\_BIZ\_C mention urgent business

body \_\_UNCLAIMED\_MONEY1 /unc/i  
body \_\_UNCLAIMED\_MONEY2 /ncl/i  
body \_\_UNCLAIMED\_MONEY3 /cla/i  
body \_\_UNCLAIMED\_MONEY4 /lai/i  
body \_\_UNCLAIMED\_MONEY5 /aim/i  
body \_\_UNCLAIMED\_MONEY6 /ime/i  
body \_\_UNCLAIMED\_MONEY7 /med/i  
body \_\_UNCLAIMED\_MONEY8 /fun/i  
body \_\_UNCLAIMED\_MONEY9 /und/i  
body \_\_UNCLAIMED\_MONEY10 /nds/i  
body \_\_UNCLAIMED\_MONEY11 /ed /i  
body \_\_UNCLAIMED\_MONEY12 /d f/i  
body \_\_UNCLAIMED\_MONEY13 / fu/i

meta UNCLAIMED\_MONEY\_C (\_\_UNCLAIMED\_MONEY1 && \_\_UNCLAIMED\_MONEY3 && \_\_UNCLAIMED\_MONEY4 && \_\_UNCLAIMED\_MONEY5 && \_\_UNCLAIMED\_MONEY6 && \_\_UNCLAIMED\_MONEY7 && \_\_UNCLAIMED\_MONEY8 && \_\_UNCLAIMED\_MONEY9 && \_\_UNCLAIMED\_MONEY10 && \_\_UNCLAIMED\_MONEY11 && \_\_UNCLAIMED\_MONEY12 && \_\_UNCLAIMED\_MONEY13)  
score UNCLAIMED\_MONEY\_C 2.7  
describe UNCLAIMED\_MONEY\_C unclaimed funds

body \_\_MILLION1 /mil/i  
body \_\_MILLION2 /ill/i  
body \_\_MILLION3 /lli/i  
body \_\_MILLION4 /lio/i  
body \_\_MILLION5 /ion/i  
body \_\_DOLLARS1 /dol/i  
body \_\_DOLLARS2 /oll/i  
body \_\_DOLLARS3 /lla/i  
body \_\_DOLLARS4 /lar/i  
body \_\_DOLLARS5 /ars/i  
body \_\_DOLLARS6 /usd/i  
meta MILLION\_DOLLARS\_C (\_\_MILLION1 && \_\_MILLION2 && \_\_MILLION3 && \_\_MILLION4 && \_\_MILLION5 && \_\_DOLLARS1 && \_\_DOLLARS2 && \_\_DOLLARS3 && \_\_DOLLARS4 && \_\_DOLLARS5 |(\_\_MILLION1 && \_\_MILLION2 && \_\_MILLION3 && \_\_MILLION4 && \_\_MILLION5 && \_\_DOLLARS6))  
#meta MILLION\_DOLLARS\_C (\_\_MILLION1 && \_\_MILLION2 && \_\_MILLION3)  
score MILLION\_DOLLARS\_C 2.5  
describe MILLION\_DOLLARS\_C mention millions of dollars

body \_\_US\_DOLLARS\_31 /(?:\\$usd)/i  
body \_\_US\_DOLLARS\_32 /[,]?\d{1,2}/i  
body \_\_US\_DOLLARS\_33 /[,]\d{2}/i  
body \_\_US\_DOLLARS\_34 /\d{3}/i  
meta US\_DOLLARS\_3\_C (\_\_US\_DOLLARS\_31 && \_\_US\_DOLLARS\_33 && \_\_US\_DOLLARS\_34)  
score US\_DOLLARS\_3\_C 2.5  
describe US\_DOLLARS\_3\_C Mentions millions of \$ (\$NN,NNN,NNN.NN)

body \_\_LOTTO\_AGENT1 /cla/i  
body \_\_LOTTO\_AGENT2 /lai/i  
body \_\_LOTTO\_AGENT3 /aim/i  
body \_\_LOTTO\_AGENT4 /ims/i  
body \_\_LOTTO\_AGENT5 /off/i  
body \_\_LOTTO\_AGENT6 /ffi/i  
body \_\_LOTTO\_AGENT7 /fic/i  
body \_\_LOTTO\_AGENT8 /ice/i  
body \_\_LOTTO\_AGENT9 /cer/i  
body \_\_LOTTO\_AGENT16 /ms /i  
body \_\_LOTTO\_AGENT17 /s o/i  
body \_\_LOTTO\_AGENT18 / of/i  
meta LOTTO\_AGENT\_C (\_\_LOTTO\_AGENT1 && \_\_LOTTO\_AGENT2 && \_\_LOTTO\_AGENT3 && \_\_LOTTO\_AGENT4 && \_\_LOTTO\_AGENT5 && \_\_LOTTO\_AGENT6 && \_\_LOTTO\_AGENT7 && \_\_LOTTO\_AGENT8 && \_\_LOTTO\_AGENT9 && \_\_LOTTO\_AGENT16 && \_\_LOTTO\_AGENT17 && \_\_LOTTO\_AGENT18)  
score LOTTO\_AGENT\_C 0.5  
describe LOTTO\_AGENT\_C claims officer

body \_\_LOTTO\_AGENT10 /fid/i  
body \_\_LOTTO\_AGENT11 /idu/i  
body \_\_LOTTO\_AGENT12 /duc/i  
body \_\_LOTTO\_AGENT13 /uci/i  
body \_\_LOTTO\_AGENT14 /cia/i  
body \_\_LOTTO\_AGENT15 /ial/i  
meta LOTTO\_AGENTBIS\_C (\_\_LOTTO\_AGENT1 && \_\_LOTTO\_AGENT2 && \_\_LOTTO\_AGENT3 && \_\_LOTTO\_AGENT4 && \_\_LOTTO\_AGENT10 && \_\_LOTTO\_AGENT11 && \_\_LOTTO\_AGENT12 && \_\_LOTTO\_AGENT13 && \_\_LOTTO\_AGENT14 && \_\_LOTTO\_AGENT15)  
score LOTTO\_AGENTBIS\_C 0.5  
describe LOTTO\_AGENTBIS\_C claims + fiducial



body \_\_WINNER1 /Win/i  
 body \_\_WINNER2 /inn/i  
 body \_\_WINNER3 /nne/i  
 body \_\_WINNER4 /ner/i  
 body \_\_WINNER5 / wi/i  
 body \_\_WINNER6 /r w/i

meta DEAR\_WINNER\_C (\_\_DEAR\_SOMETHING1 && \_\_DEAR\_SOMETHING2 && \_\_DEAR\_SOMETHING4 && \_\_WINNER1 && \_\_WINNER2 && \_\_WINNER3 && \_\_WINNER4 && \_\_WINNER5 && \_\_WINNER6)  
 score DEAR\_WINNER\_C 1  
 describe DEAR\_WINNER\_C Dear Winner

body \_\_IMPOTENCE11 /mat/i  
 body \_\_IMPOTENCE12 /rem/i  
 body \_\_IMPOTENCE13 /ema/i  
 body \_\_IMPOTENCE14 /mat/i  
 body \_\_IMPOTENCE15 /atu/i  
 body \_\_IMPOTENCE16 /tur/i  
 body \_\_IMPOTENCE17 /ure/i  
 body \_\_IMPOTENCE18 /Eja/i  
 body \_\_IMPOTENCE19 /jac/i  
 body \_\_IMPOTENCE110 /acu/i  
 body \_\_IMPOTENCE111 /cul/i  
 body \_\_IMPOTENCE112 /ula/i  
 body \_\_IMPOTENCE113 /lat/i  
 body \_\_IMPOTENCE114 /ati/i  
 body \_\_IMPOTENCE115 /tio/i  
 body \_\_IMPOTENCE116 /ion/i

#meta IMPOTENCE\_C (\_\_IMPOTENCE11 && \_\_IMPOTENCE12 && \_\_IMPOTENCE13 && \_\_IMPOTENCE14 && \_\_IMPOTENCE15 && \_\_IMPOTENCE16 && \_\_IMPOTENCE17 && \_\_IMPOTENCE18 && # \_\_IMPOTENCE19 && \_\_IMPOTENCE110 && \_\_IMPOTENCE111 && \_\_IMPOTENCE112 && \_\_IMPOTENCE113 && \_\_IMPOTENCE114 && \_\_IMPOTENCE115 && \_\_IMPOTENCE116 )

#meta IMPOTENCE\_C (\_\_IMPOTENCE11 && \_\_IMPOTENCE12 && \_\_IMPOTENCE13 && \_\_IMPOTENCE14 && \_\_IMPOTENCE15 && \_\_IMPOTENCE16 && \_\_IMPOTENCE17)  
 meta IMPOTENCE\_C (\_\_IMPOTENCE18 && \_\_IMPOTENCE19 && \_\_IMPOTENCE110 && \_\_IMPOTENCE111 && \_\_IMPOTENCE112 && \_\_IMPOTENCE113 && \_\_IMPOTENCE115 && \_\_IMPOTENCE116)  
 score IMPOTENCE\_C 2.1  
 describe IMPOTENCE\_C premature ejaculation

body \_\_BODY\_ENHANCEMENT11 /enl/i  
 body \_\_BODY\_ENHANCEMENT12 /nla/i  
 body \_\_BODY\_ENHANCEMENT13 /lar/i  
 body \_\_BODY\_ENHANCEMENT14 /arg/i  
 body \_\_BODY\_ENHANCEMENT5 /rge/i  
 body \_\_BODY\_ENHANCEMENT6 /pen/i  
 body \_\_BODY\_ENHANCEMENT7 /eni/i  
 body \_\_BODY\_ENHANCEMENT8 /nis/i

meta BODY\_ENHANCEMENT\_C (\_\_BODY\_ENHANCEMENT11 && \_\_BODY\_ENHANCEMENT12 && \_\_BODY\_ENHANCEMENT13 && \_\_BODY\_ENHANCEMENT14 && \_\_BODY\_ENHANCEMENT5 && \_\_BODY\_ENHANCEMENT6 && \_\_BODY\_ENHANCEMENT7 && \_\_BODY\_ENHANCEMENT8)  
 score BODY\_ENHANCEMENT\_C 1.6  
 describe BODY\_ENHANCEMENT\_C enlarge penis

body \_\_BODY\_ENHANCEMENT9 /siz/i  
 body \_\_BODY\_ENHANCEMENT10 /ize/i

meta BODY\_ENHANCEMENT2\_C (\_\_BODY\_ENHANCEMENT9 && \_\_BODY\_ENHANCEMENT10 && \_\_BODY\_ENHANCEMENT6 && \_\_BODY\_ENHANCEMENT7 && \_\_BODY\_ENHANCEMENT8)  
 score BODY\_ENHANCEMENT2\_C 1.5  
 describe BODY\_ENHANCEMENT2\_C penis size

```

body __GUARANTEED_100_PERCENT1 /100/i
body __GUARANTEED_100_PERCENT2 /00%/i
#body __GUARANTEED_100_PERCENTc3 /GUA/i
body __GUARANTEED_100_PERCENT4 /UAR/i
body __GUARANTEED_100_PERCENT5 /ARA/i
body __GUARANTEED_100_PERCENT6 /RAN/i
body __GUARANTEED_100_PERCENT7 /ANT/i
body __GUARANTEED_100_PERCENT8 /NTE/i
body __GUARANTEED_100_PERCENT9 /TEE/i
body __GUARANTEED_100_PERCENT10 /EED/i

body __GUARANTEED_100_PERCENT3 /gua/i

meta GUARANTEED_100_PERCENT_C (__GUARANTEED_100_PERCENT1 && __GUARANTEED_100_PERCENT2 &&
__GUARANTEED_100_PERCENT3 && __GUARANTEED_100_PERCENT4 && __GUARANTEED_100_PERCENT5 &&
__GUARANTEED_100_PERCENT6 && __GUARANTEED_100_PERCENT7 && __GUARANTEED_100_PERCENT8 &&
__GUARANTEED_100_PERCENT9 && __GUARANTEED_100_PERCENT10)
score GUARANTEED_100_PERCENT_C 2.7
describe GUARANTEED_100_PERCENT_C One hundred percent guaranteed

meta BANG_GUAR_C (__GUARANTEED_100_PERCENT3 && __GUARANTEED_100_PERCENT4 &&
__GUARANTEED_100_PERCENT5 && __GUARANTEED_100_PERCENT6 && #__GUARANTEED_100_PERCENT7 &&
__GUARANTEED_100_PERCENT8 && __GUARANTEED_100_PERCENT9 && __GUARANTEED_100_PERCENT10)
score BANG_GUAR_C 2.4
describe BANG_GUAR_C Something is emphatically guaranteed

```