Audio Hashing for Spam Detection Using the Redundant Discrete Wavelet Transform

Alexander Stoffel
Institute of Communication Engineering
Cologne University of Applied Sciences
50679 Köln, Germany
alexander.stoffel@fh-koeln.de

December 10, 2012

Abstract

For audio signals, we use the sign of the coefficients of the redundant discrete wavelet transform to generate primary hash vectors assigning bit 1 to positive or zero coefficients and bit 0 in the negative case. Discarding the highest frequency band and using a 6 step transform we get for each sample a 6 bit primary hash value which we may save as an integer. We then select a possible primary hash value (in our experiments we chose 0 or 63) and take the time indices where this primary hash value occurs as the secondary hash vector which is attributed to the whole audio signal. Comparing two audio signals, the number of elements in the intersection of the corresponding time indices are called "number of matches", a high number may indicate a similarity between the files. This secondary hash vector turns out to be robust against addition of noise, GSM-, G.726-, MP3 coding and packet loss. It may therefore be useful to detect spam telephone calls without analyzing the semantic content by the similarity of the corresponding signals. An algorithm is given to detect similar but shifted signals. Results of experiments are reported using a test corpus of 5 000 audio files of regular calls and 200 audio files of different versions of 20 original spam recordings augmented by a set of 45 files of different versions of 9 "special spam" signals.

1 Introduction

The reported results are intended to be applied in a general framework as it is described for instance in [2]. Here we focus on the calculation of audio fingerprints and their use to identify similar audio signals without analyzing their content by speech recognition. Traditionally this is done by Fourier methods. It is tempting to use wavelet methods instead, as the wavelet transform is local, first experiences are for instance reported in [1] and [3]. A further advantage of the wavelet transform is the fast algorithm to calculate wavelet coefficients by cascaded filter banks where the number of operations grows only proportionally to the number of samples (see figure 1). A disadvantage of such cascaded filter banks is the complicated behaviour of the coefficients if the signal is shifted: a shift

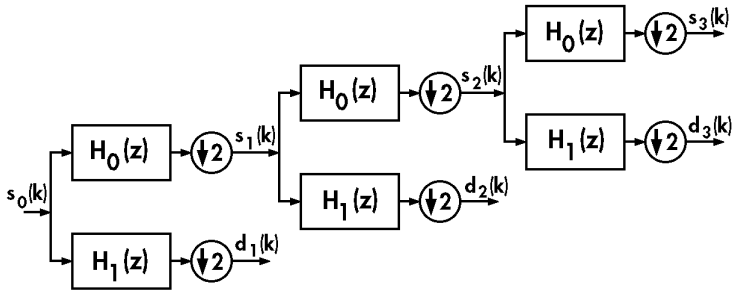

Figure 1: Cascaded filter banks to calculate wavelet coefficients

of the signal does not lead to a corresponding shift of the coefficients due to subsampling. Therefore we use here the redundant wavelet transform, which is also called stationary wavelet transform or unsubsampled wavelet transform. It does not have this disadvantage, shifted signals lead to shifted coefficients. We partially adopt normalization and notation from [5], but note that we maintain here the tilde for the analysis filters and functions. Let $\tilde{\varphi}$ be the scaling function and $\tilde{\psi}$ be the wavelet. We suppose that the wavelet is constructed in the framework of a multiresolution analysis and that the normalization is such that the low pass coefficients fulfill $\sum \tilde{h}(k) = 1$. The two scale relation is then

$$\tilde{\varphi}(x) = 2 \sum_{k \in \mathbb{Z}} \tilde{h}(k) \tilde{\varphi}(2x - k). \tag{1}$$

and the wavelet is given by

$$\tilde{\psi}(x) = 2 \sum_{k \in \mathbb{Z}} \tilde{g}(k) \tilde{\varphi}(2x - k). \tag{2}$$

The coefficients for a continuous signal f are then given by

$$r_i(k) = 2^{-i} \int f(x) \tilde{\psi}(2^{-i}x - 2^{-i}k) \mathrm{d}x, \tag{3}$$

$$s_i(k) := 2^{-i} \int f(x) \tilde{\varphi}(2^{-i}x - 2^{-i}k) \mathrm{d}x, \tag{4}$$

and may be calculated by

$$s_i(k) = \sum_{l \in \mathbb{Z}} \tilde{h}_l s_{i-1}(k + 2^{i-1}l), \quad r_i(k) = \sum_{l \in \mathbb{Z}} \tilde{g}_l s_{i-1}(k + 2^{i-1}l). \tag{5}$$

where we start as usual with $s_0(k)$ as the sampled values of the continuous signal. This may be visualized by the cascaded filter bank shown in figure 2. Replacing $\tilde{H}(z^{-1})$ by $\tilde{H}(z^{-p})$ means inserting zeros into the filter coefficients. Therefore the algorithm is frequently called "à trous" algorithm (algorithm with holes). Note that for N samples and m steps we get a $\big((m+1) \times N\big)$-matrix of coefficients of the form

$$\begin{pmatrix} r_1(1) & r_1(2) & \cdots & r_1(N) \\ r_2(1) & r_2(2) & \cdots & r_2(N) \\ \vdots & \vdots & \cdots & \vdots \\ r_m(1) & r_m(2)) & \cdots & r_m(N) \\ s_m(1) & s_m(2) & \cdots & s_m(N) \end{pmatrix} \tag{6}$$

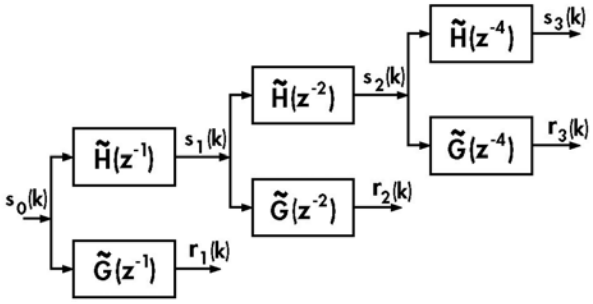

Figure 2: Cascaded filter banks for the "à trous" algorithm

As an illustration, for the interrupted pure sign shown in figure 3, the coefficients resulting from the "à trous" algorithm with 6 steps are shown in figure 4. This signal may be considered as a model for the transmission with packet loss. For the calculation of the

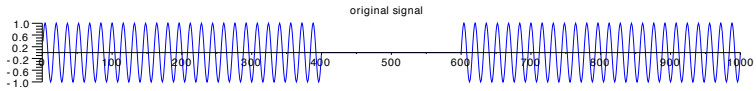

Figure 3: Original signal (pure sine with packet loss)

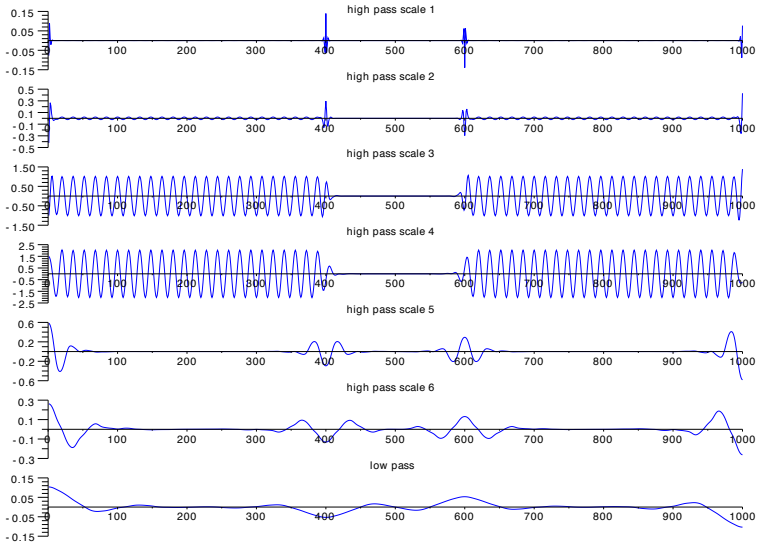

Figure 4: Redundant wavelet coefficients for the signal of figure 3

coefficients the Deslauriers-Dubuc filters DD(8,8) have been used, the implementation uses the lifting scheme with boundary conditions "reflection with repetition" in each lifting step (for the details, the reader is referred to [5]).

The same test corpus has been used as in [2]. It contains a set of 5 000 audio files obtained from regular telephone calls and a set of 200 files consisting of 20 original spam signals and 180 versions changed by packet loss, noise, audio- and telephone codecs. They are here called "normal" spam signals. This test corpus has been expanded by a set of

45 "special" spam signals which has been generated using 5 selected regular calls (by creating 8 similar versions of each). Note that if we speak of spam detection here, this means always that a similarity of the signals is detected. The semantic content is not analyzed. By its semantic content our special spam signals are regular calls.

2 First construction of a hash vector using zero crossings of wavelet coefficients

Our aim is to find a robust mapping which maps a large data vector (our audio data) to a characteristic small hash vector. Robust means that the small vector should be similar, if the audio data are changed by adding noise, amplification or other changes to hinder identification. It should also be similar if the data vector is changed by transmitting it using a noisy channel or by coding (e.g. GSM). Comparing the small hash vectors should permit to conclude that the original data are slightly changed versions of the same spam message.

The redundant wavelet transform with m steps associates to a data vector with N components a coefficient matrix with $\big((m+1)\times N\big)$ matrix elements, it increases the size of data. We therefore have to reduce them. It has turned out to be useful to neglect the high pass output of the first step, i.e. the first row of the coefficient matrix of the form (6). This reduces a part of eventually added noise. We keep only the $(m \times N)$ coefficient matrix

$$\mathbf{W} = \begin{pmatrix} r_2(1) & r_2(2) & \cdots & r_2(N) \\ \vdots & \vdots & \cdots & \vdots \\ r_m(1) & r_m(2)) & \cdots & r_m(N) \\ s_m(1) & s_m(2) & \cdots & s_m(N) \end{pmatrix} \tag{7}$$

It has been pointed out by Mallat in [4] that the original signal may be approximately reconstructed using only the zero crossings of the redundant wavelet coefficients and the sums of the coefficients between the zero crossings. This motivates the following procedure to reduce our data

$$\mathbf{W}_{ik} \mapsto \begin{cases} 1 & \text{if } \mathbf{W}_{ik} \geq 0 \\ 0 & \text{if } \mathbf{W}_{ik} < 0 \end{cases} \tag{8}$$

The zero crossings of the coefficients correspond to column indices where the value 1 changes into 0 or vice versa. We call the matrix resulting by the change (8) again $\mathbf{W}$. It may be visualized by representing a zero value by a black bar and the value one by a white bar. This is done in figure 5 for a part of the original spam signal "SPIT_03_Behring" and its GSM-coded version. It is clearly visible that the obtained binary values reflect well the similarity of the signals.

In our experiments, we choose a wavelet transform with 6 steps in scale. Discarding the finest scale, the above procedure furnishes therefore 6 binary values per sample. We take the corresponding integer and obtain an integer hash value between 0 and 63 for each *sample*. We shall call this integer here the *primary* hash *value*.

For each file we neglect the first 10 000 audio samples, perform the wavelet transform for the following 50 400 samples. To avoid the influence of boundary conditions of the wavelet transform, we omit all coefficients belonging to the first and last 200 samples. We calculate the primary hash values for the remaining samples and obtain for each *file* a *primary* hash *vector* with 50 000 integer components. At a sampling frequency of 8 kHz

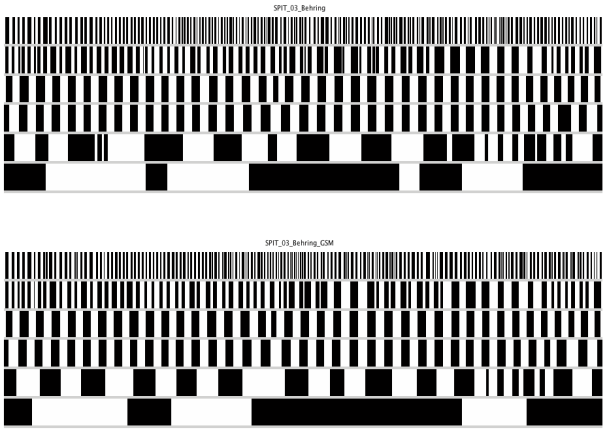

Figure 5: Visualistion of the binary values obtained from the wavelet coe cients by (8) for a spam signal (above) and its GSM-coded version (below)

this means that we use only a part of a duration of 6.25 s of our signal. As we discard the nest scale, we only investigate a frequency range below 2 kHz. Performing 6 steps in scale means that the high pass coe cients of the coarsest scale we use correspond essentially to a frequency range of 60 Hz.

Our procedure still generates far too much data for practical purposes. But it is illustrative to look at the results of a rst test obtained by calculating the Hamming distances of the generated vectors for all 200 normal spam audio les of our test corpus. By symmetry it is su cient to calculate the Hamming distances $d(i\ k)$ between le i and k only for the case $i > k$, i.e. the part below the diagonal of the corresponding matrix. In gure 6 each matrix element below the diagonal is represented by a gray value, the maximal value corresponding to white, the minimal value to black. The dark triangles

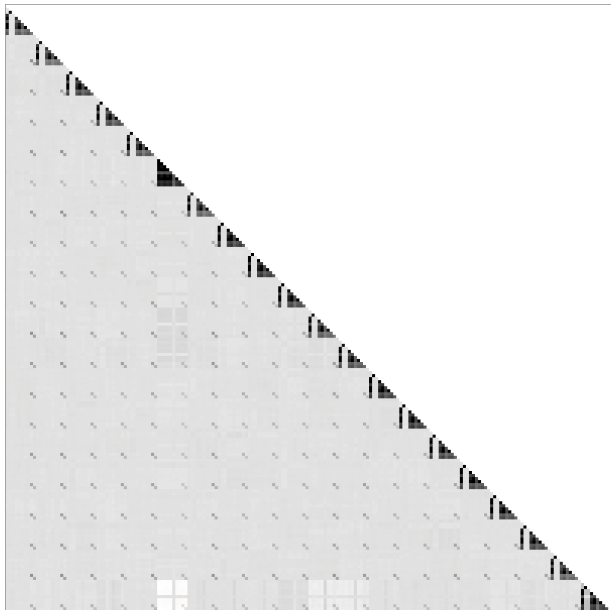

Figure 6: Hamming distances for the binary hash values obtained by (8), the maximal value corresponds to white, the minimal to black

clearly show that most of the di erent versions of the same original spam le furnish hash vectors which have a relatively low Hamming distance. This indicates that we may recognize most spam signals by calculating the Hamming distance of our hash vectors. But a closer look to the triangles shows that they all have gaps in the form of two bright columns and two bright pixel in the upper left part. This is more clearly visible at gure 7 (which is a zoomed version of the upper left part of gure 6).

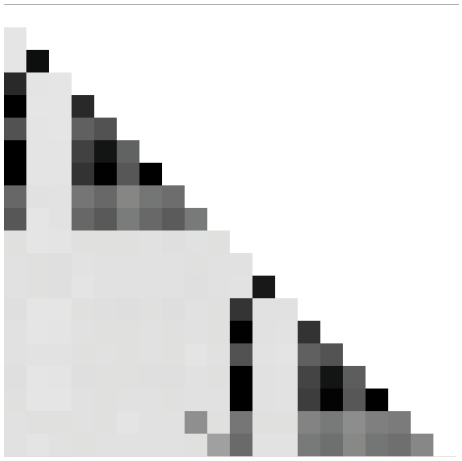

Figure 7: Hamming distances for the di erent versions of the spam les of type SPIT_01_ABeier and SPIT_02_ABeier2 (zoom of the upper left part of gure 6)

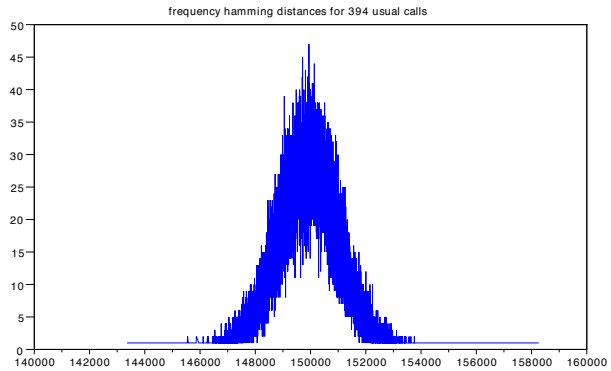

Figure 8: Frequency of the values of the Hamming distance for 77 421 pairs of regular calls

The bright columns and the two bright pixel in the upper left part of the triangles correspond to pairs which contain the versions 32kbps and 96kbps . They are generated by a shift and a subsequent MP3 coding. This shift has has not yet been taken into account here and explains the high value of the Hamming distance despite the similarity of the signals.

The visualization of the Hamming distances in gure 6 shows quite amazing pairs of dark pixel spread over the whole range and outside the triangles where the similarity of the signals explains small distances. It turns out that they belong to pairs of the noised versions of di erent spam signals, but with the same type of noise, for instance to the pairs (SPIT_01_ABeier_n20w, SPIT_02_ABeier2_n20w) and (SPIT_01_ABeier_n20p,

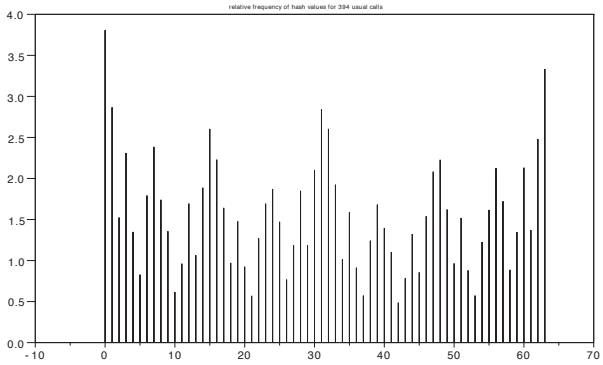

Figure 9: Relative frequency (in percent) of the hash values for 394 regular calls with 50 000 samples each

SPIT_02_ABeier2_n20p)". Those pairs are also visible in figure 7 at the left of the lower left corner of the lower triangle. The similarity of the corresponding signals may be explained by the fact that the same calculated noise has been added to different original signals. It is remarkable that the hash vectors calculated here make this visible by a smaller Hamming distance.

The behaviour of the Hamming distances of our hash vectors for the regular telephone signals has been tested also. 400 files of our corpus have been randomly chosen. 6 of them turned out to be too short to get the 60 400 samples we use, those too short files have been discarded. From the remaining 394 files we get 77 421 different pairs for which the Hamming distances have been calculated. Figure 8 shows the frequency of the different values of the Hamming distances. We get a mean value $\mu = 149\,916.65$ and a standard deviation of $\sigma = 1\,108.005$. Supposing that the hash vectors are completely random we would have an expectation value of $\mu = 150\,000$ and a standard deviation of $\sigma = 273.861$. The mean value agrees quite well, but the empirical standard deviation is larger than the value resulting from randomness. In fact, our hash vectors are not completely random as their relative frequency shows for 400 at random chosen regular calls in figure 9. The hash values 0 and 63 are far more frequent than for instance 21 and 42.

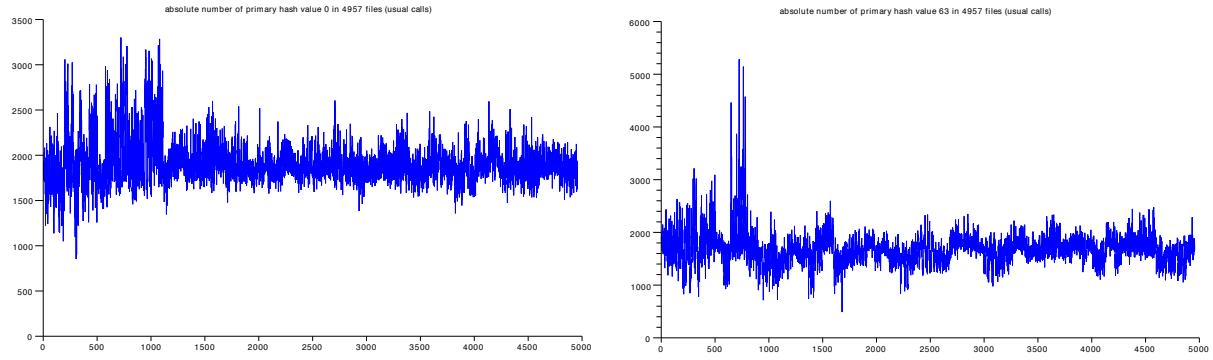

Figure 10: Occurrences of primary hash value 0 (left) and 63 (right) in 4957 files for regular calls

The number of of occurrences of the primary hash values 0 and 63 has been tested for the whole corpus of 5000 regular calls, the result for 4957 files (the remaining do not

contain enough samples) is shown in figure 10. Apparently about the first 1000 files behave differently with respect to the chosen primary hash values. This is a further indication that the occurrences of those values may not be considered as random.

3 Index positions of a primary hash value as new hash vector

We denote our primary hash vector for each file $\mathbf{h} = (h_1, h_2, h_3, \ldots h_N)$ where $h_k \in \{0, 1, 2, \ldots 63\}$ and N is the number of samples taken from the file (remember that we chose here $N = 50\,000$). This hash vector would furnish far too much data, and the calculation of the Hamming distances would be too slow in practice. We have therefore to reduce further those data. For this purpose, we select one fixed possible primary hash value $v \in \{0, 1, 2, \ldots 63\}$ and keep the index positions of this value in the vector $\mathbf{h}$ as new *secondary* hash vector $\mathbf{t}$, i.e. we consider the set

$$H_v = \big\{k \in \{1, 2, \ldots N\} \mid h_k = v\big\} \tag{9}$$

and take for each file the elements of H_v in increasing order as the components of the *secondary* hash vector

$$\mathbf{t} = (t_1, t_2, t_3, \ldots t_L) \quad \text{with} \quad t_k \in H_v \quad \text{and} \quad 1 \le t_1 < t_2 < t_3 < \ldots t_L \le N \tag{10}$$

Note that the number of components L of this vector $\mathbf{t}$ differs from file to file (see figure 10). For our test corpus and $v = 0$ or $v = 63$ a typical value is 2000. The use of the secondary hash vector reduces our data from 300 000 to typically 32 000 bit per file.

As a measure of similarity of two files we take the number of elements in the intersection of the set of components of the corresponding secondary hash vectors, i.e. we consider for two secondary hash vectors

$$\mathbf{s} = (s_1, s_2, s_3, \ldots s_L) \quad \text{and} \quad \mathbf{t} = (t_1, t_2, t_3, \ldots t_M)$$

the number

$$n_m := \Big|\{s_1, s_2, s_3, \ldots s_L\} \cap \{t_1, t_2, t_3, \ldots t_M\}\Big| \tag{11}$$

Adopting our language to the procedure in [2] we shall call those index values in the intersection "matches". They indicate the time indices where the primary hash values in both files agree with the chosen value (e.g. 0 or 63). Note that a high number of matches n_m signifies a great similarity, a low number a small similarity. We expect that even for regular calls we may have a small number of matches. Instead of the clumsy notation of (11) we shall use here the shorthand notation

$$n_m := \big|\mathbf{s} \cap \mathbf{t}\big| \tag{12}$$

Figure 11 shows the result of two experiments with regular calls. For the left side we choose the primary hash value 0 and for the right side the primary hash value 63. The relative frequency of the number of matches for all possible pairs is shown (in black). Furthermore the corresponding normal distribution is indicated (in blue) and the corresponding gamma distribution in red. The calculations are based in both cases on the random selection of 2 000 files where 20 files turned out to be too short. Therefore the

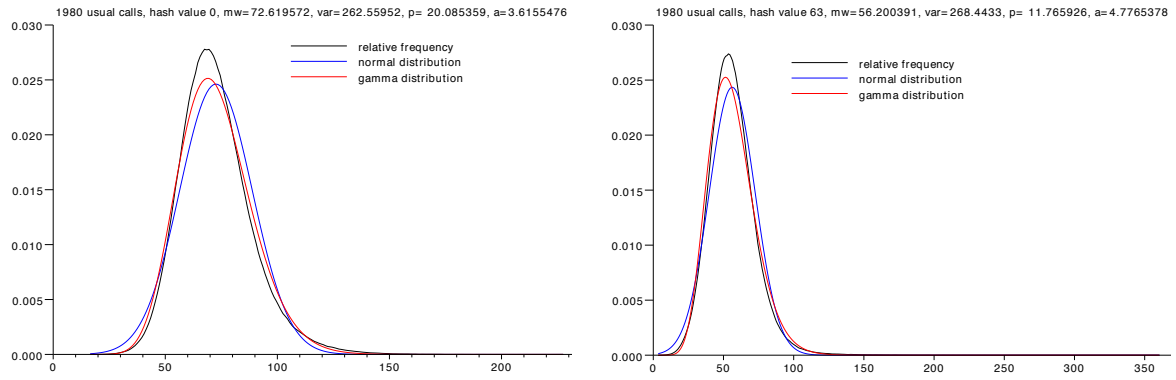

Figure 11: Relative frequency of matches for regular calls and primary hash values 0 (left) and 63 (right)

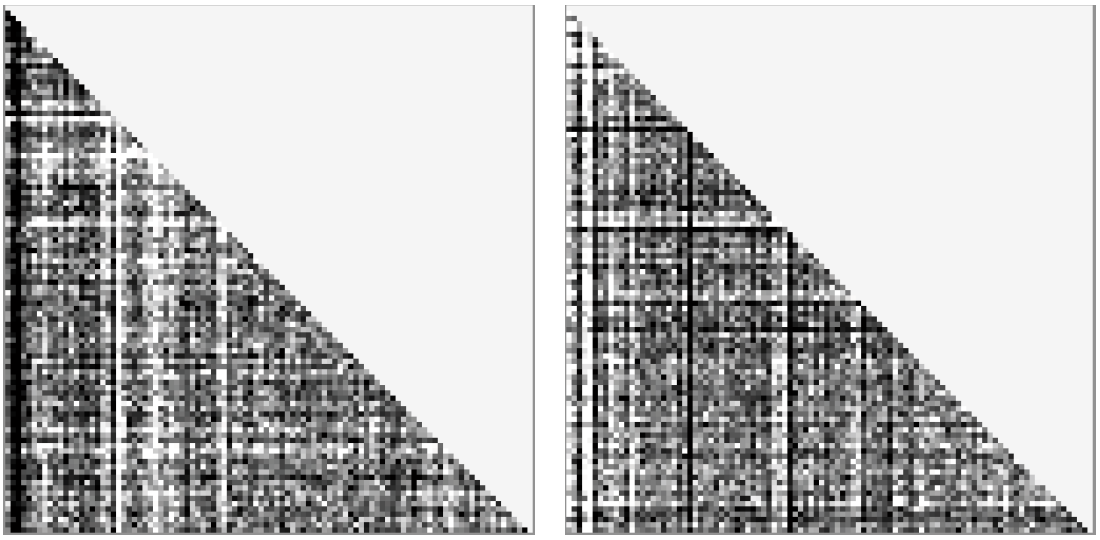

Figure 12: Number of matches for regular calls and primary hash values 0 (left) and 63 (right) for 100 randomly chosen les. White corresponds to a number greater or equal to + 1 5 , black to less or equal to 1 5

number of matches had to be calculated for 1 959 210 pairs. The parameters for the gamma distribution have been calculated from the empirical mean and variance. The gamma distribution seems to be the better approximation to the experimental one, but the asymptotic decrease of the experimental distribution is apparently even slower.

Figure 11 might give the impression that the number of matches for regular calls is completely random. In gure 12 the number of matches is visualised for experiments with only 100 les. The dark and bright lines and columns clearly indicate that for certain les all pairs containing this le have a particular high or low number of matches. This may be explained by the di erent number of the selected primary hash value in certain les. A very large number of elements in one set may lead to a larger number of elements in the intersection with many other sets.

Considering the statistics of the number of matches for regular calls in gure 11 a reasonable rst trial for a threshold for spam is 200, i.e. we may consider two signals as

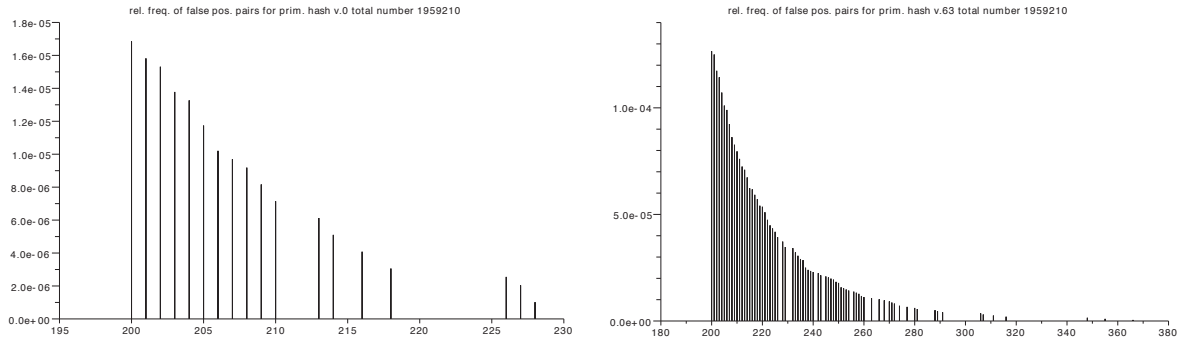

Figure 13: Decrease of the relative frequency of false positive diagnostics if the threshold is increased for primary hash value 0 (left) and 63 (right), a total number of more than $1.9 \cdot 10^6$ pairs have been analyzed

similar and therefore classify them as spam if the number of matches for 0 or 63 is above or equal to 200. Supposing that the number of matches for regular calls is given by the gamma distribution, this threshold for spam detection would lead to a probability for a false positive classification of $1.9 \cdot 10^{-8}$ for the primary hash value 0 and a probability of $2.6 \cdot 10^{-8}$ for the primary hash value 63. Note that this is an underestimation as the experimental values for large number of matches lie above those of the gamma distribution. In fact, we had false positive diagnostics for pairs of regular calls with the chosen threshold. We might reduce their number by increasing the threshold, but this would increase the number of undetected spam pairs. The decrease of the frequency of false positive pairs depending on the threshold is shown in figure 13, this is based on an experiment with more than $1.9 \cdot 10^6$ pairs of files. Note that those false positive results do not seam completely random. It is quite amazing that all the concerned filenames are within a relatively small part of the alphabetic list of the filenames (for the primary hash value 0 see table 3 in the appendix).

It should be remarked that we have as quite exceptional result 1653 matches for the files "g215aB5" and "g218aB6" for the hash value 63. The number of matches for more then $12 \cdot 10^6$ pairs has been calculated to find this exception. This pair does not belong to those randomly chosen for figure 11. This event would be highly improbable if the number of matches were really gamma distributed.

Despite those reflections we used the threshold 200 for our experiments. Figure 14 shows the corresponding result for our normal test spam signals, all pairs of files identified as similar in that sense correspond to a white pixel, values below the threshold are darker, black corresponds to the minimal value. Qualitatively this result looks very similar to the Hamming distances in figure 6 if one notes that here darker gray corresponds to less similar signals. Again the different spam sorts are visible as triangles with some gaps — at the same place for different sorts — which correspond to shifted signals not recognized as similar. And again most of the files of different spam type with the same kind of calculated noise are recognized as similar.

But there are some remarkable differences to the Hamming distances: For the sixth type of spam (characterized by "SPIT_06_Bank" in the file name) even the versions containing a shift are recognized as similar. For the primary hash value 0 none of the spam versions of the last type (characterized by "SPIT_20_unbek" in the file name) would be recognized as similar. Furthermore the number of matches for pairs of different spam

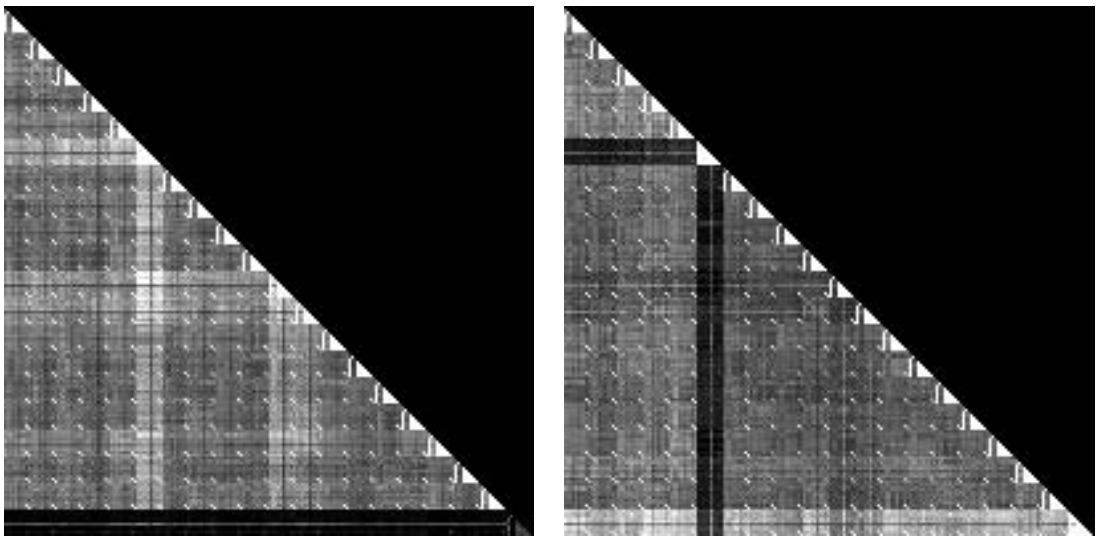

Figure 14: Number of matches for normal spam calls and primary hash value 0 (left) and 63 (right), pairs with matches above the threshold of 200 correspond to white pixel, smaller values correspond to darker gray values

types should be quite low, but this is not the case. Furthermore, pairs containing spam type 6 signals have a comparatively large number of matches for primary hash 0 and an exceptionally low number of matches for primary hash 63 (clearly visible as thick black "column" and "line"). And for spam type 20 (the last one), the opposite may be stated. We have a comparatively large number of matches for 63 and a remarkably low number of hashes for 0.

Therefore the statistics of matches of pairs of different normal spam types and of pairs formed by regular calls and the original spam messages has been investigated for the selected primary hash values 0 and 63, the result is shown in figure 15. It is qualitatively different from the corresponding results obtained by pairs of regular calls shown in figure 11. A striking difference is the appearance of a second local maximum. Those results indicate that the normal spam signals have some qualitative differences to the signals of the regular calls. Great care is therefore needed not to base the spam detection on such differences which might be for instance due to different techniques of registration. The number of occurrences of the selected primary hash values 0 and 63 for the original spam signals in figure 16 shows some further particularity. For the last type, the hash value 0 occurs only 90 times, but the occurences of 63 attain their maximum with 3526. This explains why this type of spam may not be detected using the primary hash value 0, because the intersection with a set of 90 indices may never furnish enough elements to attain the threshold of 200.

It might be tempting to recommend the primary hash value 63 instead of zero. But this would be based on some unknown property of the last spam type, and there might arise some new type of spam for which there are problems for the detection with this primary hash value. Perhaps it is preferable to work with both selected primary hash values 0 and 63. It should also be noted that this strange problem to detect the last spam type using the primary hash value 0 disappears using the offset correction discussed in section 6.

It has been considered that it might be necessary to test if our method depends on

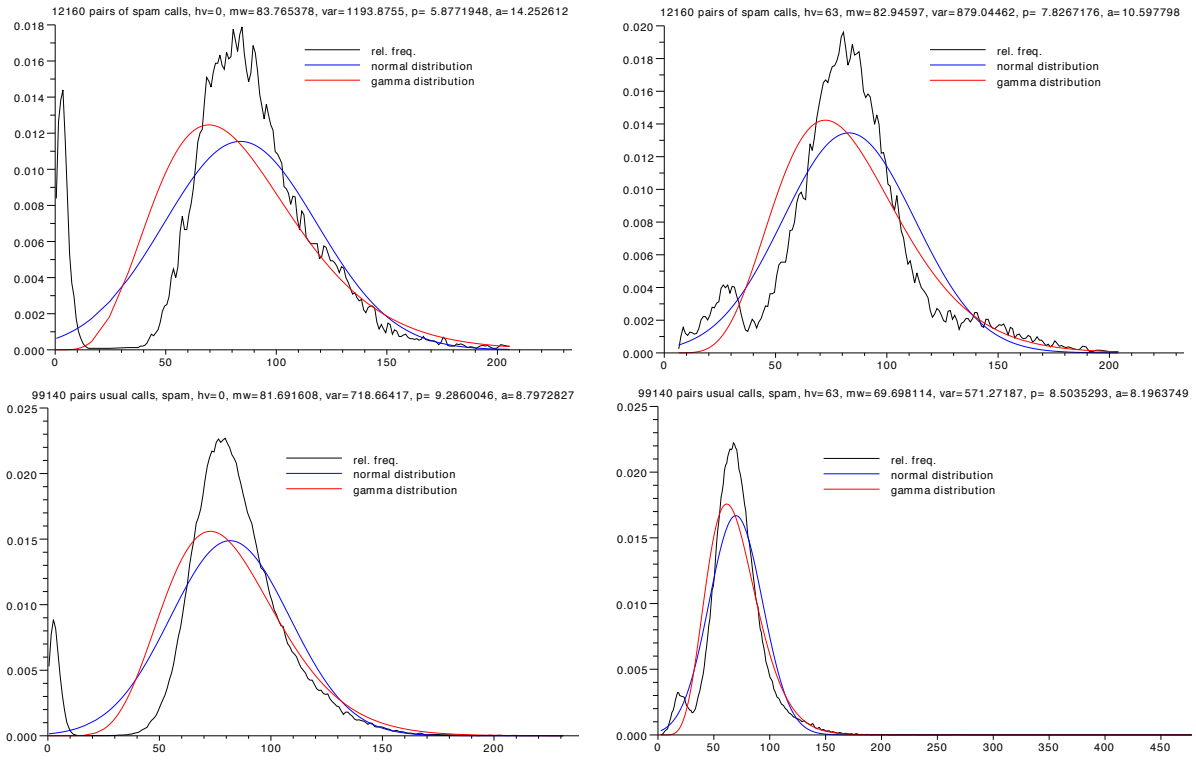

Figure 15: Relative frequency of matches for pairs of all versions of different normal spam calls (above) and for pairs formed by regular calls and original spam signals (below). The graphics at left show the results for the primary hash value 0, at right for 63

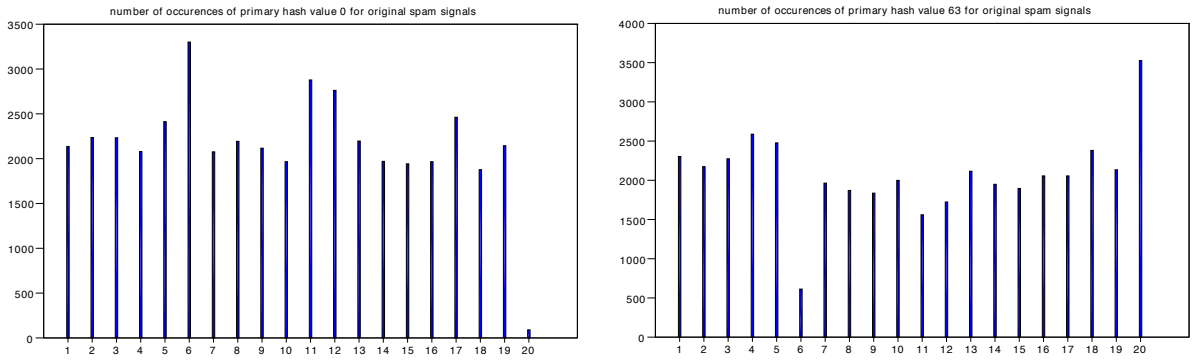

Figure 16: Number of occurrences of primary hash value 0 (left) and 63 (right) for the original spam signals

some unknown particularity of our normal spam test files. Therefore 5 regular calls have been chosen to produce a special spam test corpus by creating similar versions. White and pink noise has been added, the same type of packet loss has been simulated, two versions of shifted and MP3-coded signals and two versions of G.726-coded have been created to obtain 9 versions for each special spam file (compared to the "normal" spam files the GSM version is lacking). It should be remarked that the files of regular calls chosen to produce the special spam files have been selected to give a low number of primary hash value 63 for the variant discussed in section 5 (low energy elimination). Here (i.e. without

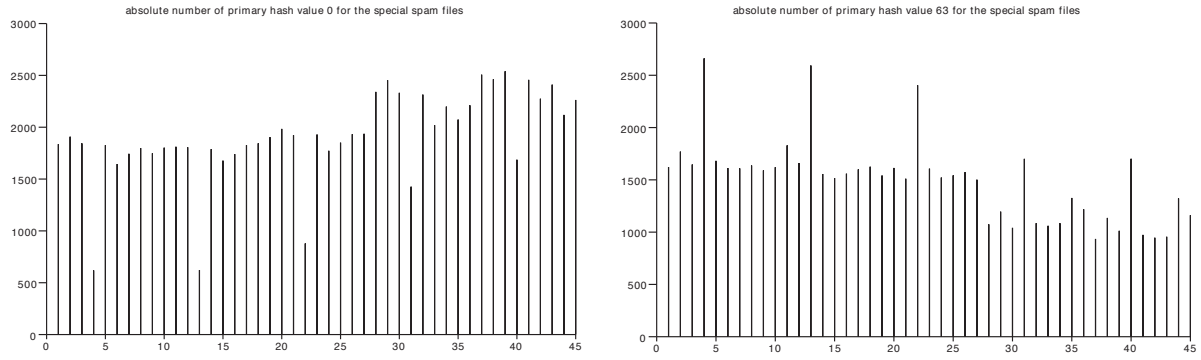

Figure 17: Number of occurrences of primary hash value 0 (left) and 63 (right) for all special spam signals

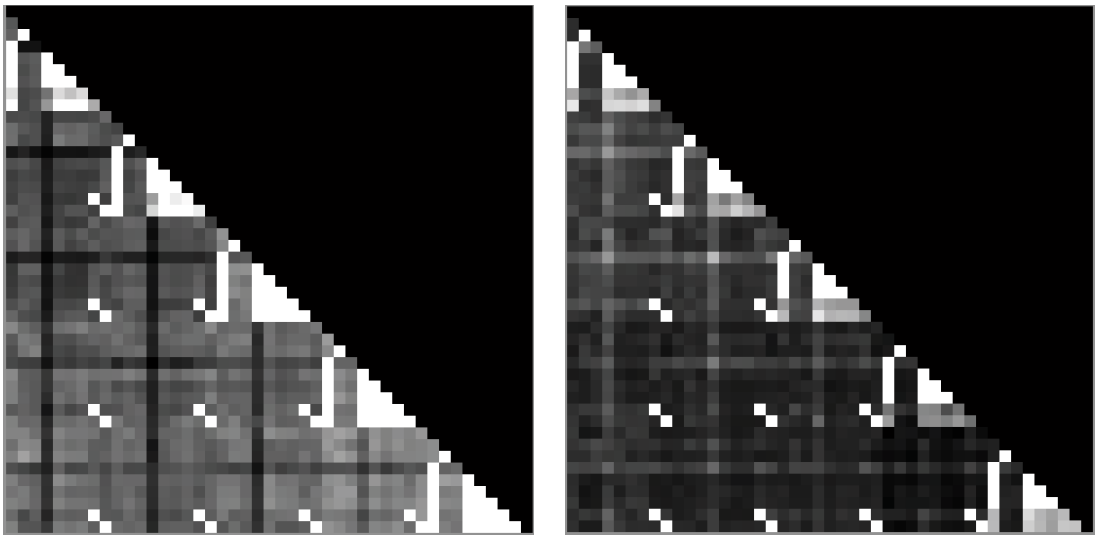

Figure 18: Number of matches for the special spam calls (generated by 5 regular calls) for primary hash value 0 (left) and 63 (right), pairs with matches above the threshold of 200 correspond to white pixel, smaller values correspond to darker gray values

the low energy elimination), the number of occurrences of the primary hash values shown in gure 17 does not show any particularity except for the G.726-coded versions (le numbers 4, 13, 22, 31 and 40). The number of matches for all special spam les and the selected primary hash values is visualized in gure 18. Comparing to gure 14 we see again the light triangles corresponding to detected spam pairs which contain some darker parts corresponding to problem pairs such as the les with a shifted signal. Furthermore, for primary hash value 63 there seems to be a problem for all types of special spam to detect the similarity with noisy signals of the same spam type (n20p and n20w , indicated by the two darker horizontal lines at the bottom of the triangles), although the number of matches is not much below the threshold. But all pairs of di erent spam type with the same kind of calculated noise are recognized as similar.

It might seem a good idea to modify our criterion for spam detection taking some normalization of the number of intersection of the indices where the selected primary hash value occurs. A possibility would be to divide by the product or the geometric mean

of the number of elements of both sets of indices. But first essays were discouraging. One reason is possibly that the quotient of two random variables may have undesirable properties as has the the quotient of two normally distributed variables (the probability density function is very slowly decaying for large values).

Furthermore it should be stated that many parameters in the experiments presented here have been chosen by trial and error. Some caution is needed to make sure that the choice of the parameters does not represent a local optimum and there might be a completely different and far better choice of the parameters. Therefore a careful study of the influence of the parameters is to be recommended.

4 Detecting similar but shifted signals using matches of the secondary hash vector

We have to tackle the detection of shifted versions of spam signals as the versions of type "32kbps" and "96kbps". We fix first the maximal number of shifts we want to detect as Δ_m, i.e. we want to detect signals as similar where the samples have been shifted by a number of index positions up to $\pm\Delta_m$. For our experiments, we fixed $\Delta_m = 2000$ which corresponds with a sampling frequency of 8 000 Hz to a time shift of 0.125 seconds. An easy but very time consuming method consists in testing all possible shifts, i.e. to consider the maximal number of matches for all shifts in this range, i.e.

$$n_s := \max_{-\Delta_m \le k \le +\Delta_m} \Big| \{s_1 + k, s_2 + k, s_3 + k, \dots s_L + k\} \cap \{t_1, t_2, t_3, \dots t_M\} \Big| \text{ with } k \in \mathbb{Z} \quad (13)$$

or using the shorthand notation of (12)

$$n_s := \max_{-\Delta_m \le k \le +\Delta_m} \big| (\mathbf{s} + k) \cap \mathbf{t} \big| \quad (14)$$

where the sum of a vector $\mathbf{s}$ and an integer scalar k is understood here componentwise as indicated in the detailed notation of (13). If n_s is greater than or equal to the threshold, we may qualify the corresponding signals as spam. Note that this method is based on the fact that we use here the redundant wavelet transform which has a simple behaviour under translations: the shifted signal has correspondingly shifted coefficients and therefore leads to a shifted secondary hash vector.

To get a faster detection of shifted signals we proceed here in three steps. First we check on a coarse scale if one signal might be the shifted version of a signal similar to the other one. We consider a coarse approximation to the procedure in (13) by dividing the maximal shift Δ_m in m_c subshifts $\Delta_m = m_c \cdot \Delta_c$. For this coarse approximation we use the coarse approximate hash vector

$$\mathbf{s}_c := \left(\left\lfloor \tfrac{s_1}{\Delta_c} + \tfrac{1}{2} \right\rfloor, \left\lfloor \tfrac{s_2}{\Delta_c} + \tfrac{1}{2} \right\rfloor, \dots \left\lfloor \tfrac{s_L}{\Delta_c} + \tfrac{1}{2} \right\rfloor \right) \quad (15)$$

and an analogously defined approximate hash vector $\mathbf{t}_c$ to construct the approximate number of matches

$$n_c = \max_{-m_c \le k \le +m_c} \big| \mathbf{s}_c + k \cap \mathbf{t}_c \big| \quad (16)$$

Note that this approximation means that we use an approximate time scale by dividing the secondary hash values (which are time indices) by Δ_c and rounding to integer values. Using this coarse time scale we need only to check translations up to $\frac{\Delta_m}{\Delta_c}$. In our experiments we used $\Delta_c = 80$ and $m_c = 25$.

If the number of matches obtained by (16) is greater than or equal to the threshold, we call k_c the shift k where the maximum is attained. We expect that this value is unique if $\mathbf{s}$ comes from a shifted and slightly modified version of the signal which generates $\mathbf{t}$, otherwise we take the rounded mean of all the values where the maximum is attained. In the original scale we expect that a shift of approximately $k_c \cdot \Delta_c$ will give a maximum of matches in (14), i.e. we may hope the shift we are looking for is in the interval $[(k_c - 1) \cdot \Delta_c, (k_c + 1) \cdot \Delta_c]$.

To improve this estimation we consider an intermediate scale dividing the maximal shift Δ_m in a larger number of smaller subshifts Δ_i with $\Delta_i < \Delta_c$. We construct approximate hash vectors $\mathbf{s}_i$ exactly as in (15) with Δ_c replaced by Δ_i. But now we need not try the whole range because we have already a good starting point by the interval found by the search on the coarse scale. We only have to rescale this interval to the actual intermediate time scale as here all time indices are divided by Δ_i and rounded. We therefore get from the "coarse" interval the condition for the intermediate indices

$$(k_c - 1)\frac{\Delta_c}{\Delta_i} \leq k \leq (k_c + 1)\frac{\Delta_c}{\Delta_i}$$

Introducing the abbreviation $k_s := k_c \cdot \frac{\Delta_c}{\Delta_i}$ we get the following intermediate approximation

$$n_i = \max_{k_s - \frac{\Delta_c}{\Delta_i} \leq k \leq k_s + \frac{\Delta_c}{\Delta_i}} \left| \mathbf{s}_i + k \cap \mathbf{t}_i \right| \tag{17}$$

The index k where this maximum is attained (or if this is not unique the rounded mean of the corresponding indices) is now designated by k_i and this furnishes the smaller interval $[(k_i - 1) \cdot \Delta_i, (k_i + 1) \cdot \Delta_i]$. We hope our shift index is in this interval.

To get the final value, we have only to determine for $k_f := k_i \cdot \Delta_i$

$$n_f = \max_{k_f - \Delta_i \leq k \leq k_f + \Delta_i} \left| \mathbf{s} + k \cap \mathbf{t} \right| \tag{18}$$

using our original secondary hash vectors $\mathbf{s}$ and $\mathbf{t}$. This value will be taken as the final number of matches for the decision if the investigated pair of signals is spam.

Note that for the simple trial in (14) we need $2 \cdot \Delta_m + 1$ calculations of intersections, for our approximation in 3 steps we need only

$$2 \cdot m_c + 1 + 2 \cdot \frac{\Delta_c}{\Delta_i} + 1 + 2\Delta_i + 1$$

calculations. For our experiments this is a reduction from 4001 to 89 calculations of intersections.

The result of an analysis of all normal spam versions of our test corpus with the above described shift correction using the primary hash values 0 and 63 is shown in figure 19. White pixel belong to a number of matches above the threshold which is again 200. The corresponding pair is classified as spam. The success of our treatment is visible comparing with the result without shift correction in figure 14. The dark columns inside and the two dark squares at the top left of the white rectangles belonging to the shifted signals of type "32kbps" and "96kbps" disappeared, the shifted signals are nearly all detected as similar. Only a few isolated gray pixel inside the white triangles indicate some non detected spam pairs as remaining problem cases — and again the completely undetected spam type in the case of primary hash value 0 is visible at the lower right corner. A complete list of

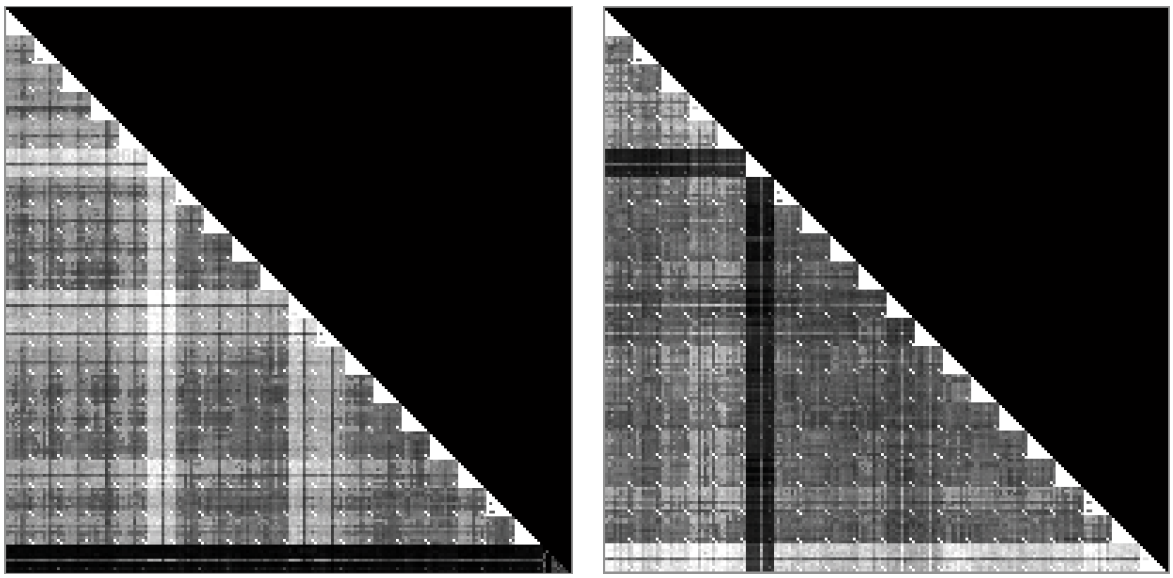

Figure 19: Matches for normal spam les with detection of shifted signals for primary hash value 0 (left) and primary hash value 63 (right). A white pixel indicates a number of matches at or above the threshold, therefore detection of the corresponding pair as spam

the remaining undetected spam signal pairs is given in table 5 and 6 (in the appendix). Note that the whole test corpus has 20 types with 200 normal spam signals. This means that a total number of 900 spam pairs should be detected. We have only about 6 2 % problem pairs for primary hash value 0 and 1 4 % problem pairs for primary hash value 63. As a result we get that the shift detection presented here fails only with very few spam versions of type n20p (pink noise) and GSM .

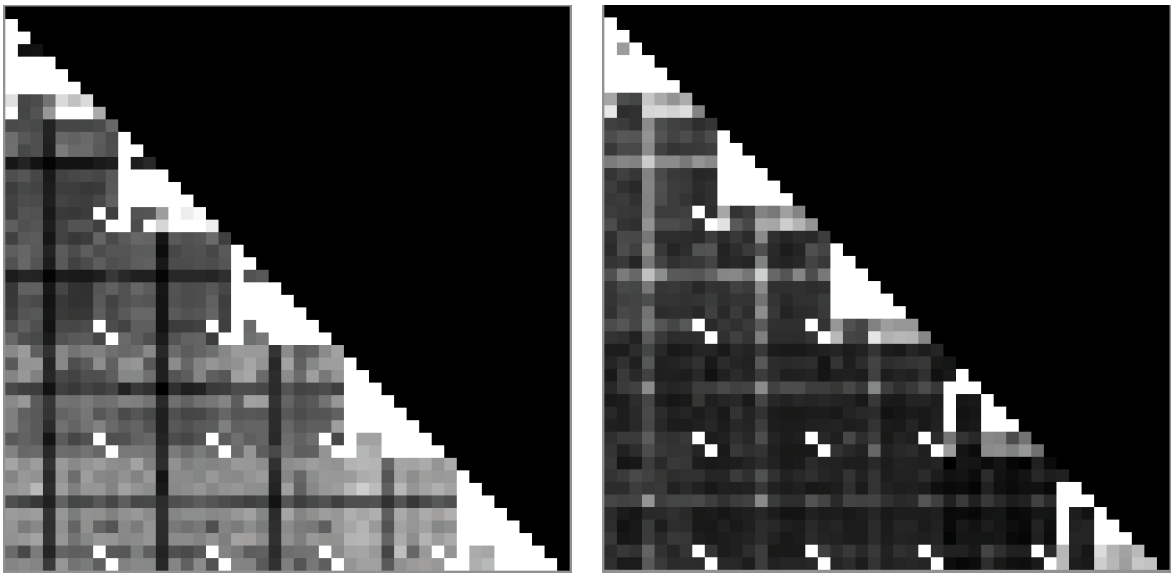

Figure 20: Matches for special spam les with detection of shifted signals for primary hash value 0 (left) and primary hash value 63 (right). A white pixel indicates a number of matches at or above the threshold, therefore detection of the corresponding pair as spam

Again our method has been tested also for the special spam files generated by selected regular calls, the result is shown in figure 20. Some of the noisy versions still lead to problems with spam detection especially if the other file contains a shift. For the primary hash value 0 we get 36 undetected spam pairs (20 %, as we should detect 180), and for the primary hash value 63, there are two exceptional spam types (the last in our ordering), where most of the shifted signals are not recognized. We get a total number of 91 undetected pairs (51 %).

5 Elimination of signal parts with low energy

A lot of variants of the procedure described here are possible. We shall restrict ourselves to very few of them, one being the elimination of signal parts with low energy. This means in practice that we may decide not to consider moments of silence. Here, to calculate the energy, we use only the wavelet coefficients and not the original signal. Note that therefore even very small amplitudes may lead to significant signal energy if their are abrupt changes. As a measure for signal energy, for each sample the Euclidean norm of the corresponding column vector of the matrix $\mathbf{W}$ given by (7) is calculated. Let us call $\mathbf{w}_k$ the kth column vector of $\mathbf{W}$. We fix a threshold $\varepsilon > 0$ and discard the corresponding column index in the calculation of the secondary hash vector, if the norm of the column vector is not above the threshold. I.e. we replace (9) by

$$H_v = \big\{k \in \{1, 2, \dots N\} \mid h_k = v \text{ and } \|\mathbf{w}_k\| > \varepsilon\big\} \tag{19}$$

Note that the secondary hash vector $\mathbf{t}$ formed by the elements of H_v in increasing order then has only components where the signal energy is above the threshold, in general it will become shorter by this additional condition.

It is remarkable that this procedure may essentially modify the number of matches for pairs of signals. This is particularly striking for the exceptional pair "g215aB5" and "g218aB6" leading to a false spam result with 1653 matches. Table 1 shows how this number of matches decreases dramatically within a small range of the threshold ε. The number of samples above the threshold N_k does not decrease very much in this range and even the length of the corresponding secondary hash vectors decreases only by a factor $\frac{1}{2}$, whereas the number of matches decreases by a factor $\frac{1}{10}$.

In contrast to this result the statistics of primary hash values as shown in figure 9 is only slightly changed by the elimination of the low energy parts. This can be seen in figure 21 for the threshold values of $\varepsilon = 10^{-5}$ and $\varepsilon = 10^{-3}$. No dramatic change with respect to figure 9 is visible, there is a slight increase of the occurrences of the preferred values 0 and 63.

But the absolute value of the number of occurrences for our preferred primary hash values decreases much as is shown in figure 22 (to be compared with figure 10). This might prevent us in many cases to use the number of matches for spam detection, as the number of elements of the intersection may not be larger than the number of elements in the concerned single sets. We should therefore have a look at the statistics of the matches for the regular calls in figure 23. The number of matches has decreased by the elimination of low energy samples, but it should be noted that there have been found some few pairs with number of matches above 150 and even one pair with 182 matches for the hash value 0 and 203 matches for the hash value 63. In a further experiment with more than $1.9 \cdot 10^6$ pairs randomly chosen pairs, for the primary hash value 0 there has been found

ε	N_1	N_2	L_1	L_2	n_m
1.00000D-09	50000	50000	5284	3841	1653
5.00000D-08	49963	47505	5284	3784	1651
5.50000D-08	49618	46806	5270	3750	1647
5.80000D-08	49359	46318	5230	3708	1640
5.85000D-08	49318	46231	5228	3701	1640
5.90000D-08	49085	46065	5219	3696	1637
5.95000D-08	48667	45763	5155	3655	1628
5.96000D-08	48292	45558	4963	3540	1593
5.96040D-08	48102	45442	4851	3482	1567
5.96045D-08	48079	45405	4851	3476	1566
5.96046D-08	48050	45389	4846	3468	1556
5.96047D-08	45725	43673	2670	1948	115
5.96050D-08	45620	43538	2618	1858	108
6.00000D-08	44701	43038	2260	1703	88
1.00000D-07	42733	37975	2121	1526	76
1.00000D-06	39048	17408	2113	799	43
1.00000D-05	29674	11577	2032	763	42

Table 1: Signal energy threshold ε, number of samples N_k above threshold, number of components L_k of the corresponding secondary hash vector for primary hash value 63 for the exception files "g215aB5" (index 1) and "g218aB6" (index 2) and number n_m of resulting matches

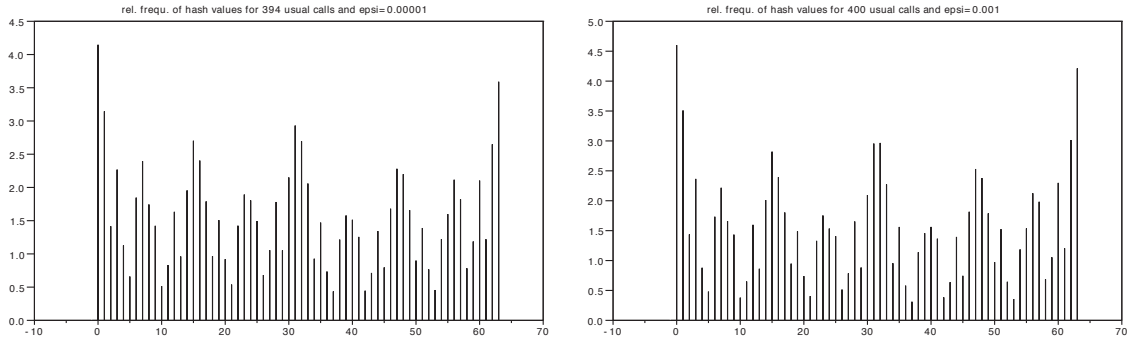

Figure 21: Relative frequency of the primary hash values with elimination of low energy parts with threshold $\varepsilon = 10^{-5}$ (left) and $\varepsilon = 10^{-3}$ (right)

one pair of regular calls just at the threshold 200 and none above and none at or above the threshold for the primary hash value 63. And in a third experiment there has been no pair at or above the threshold for both primary hash values. Lowering the threshold for the number of matches for spam detection would not be wise and is — as it turns out — not necessary.

Despite of the decrease of the number of occurrences of the primary hash values, using low energy elimination with threshold $\varepsilon = 10^{-5}$ does not cause a significant change for the detection of our normal spam signals, and this minor change is in opposite direction for the two primary hash values 0 and 63. This is shown by the list of undetected spam pairs in

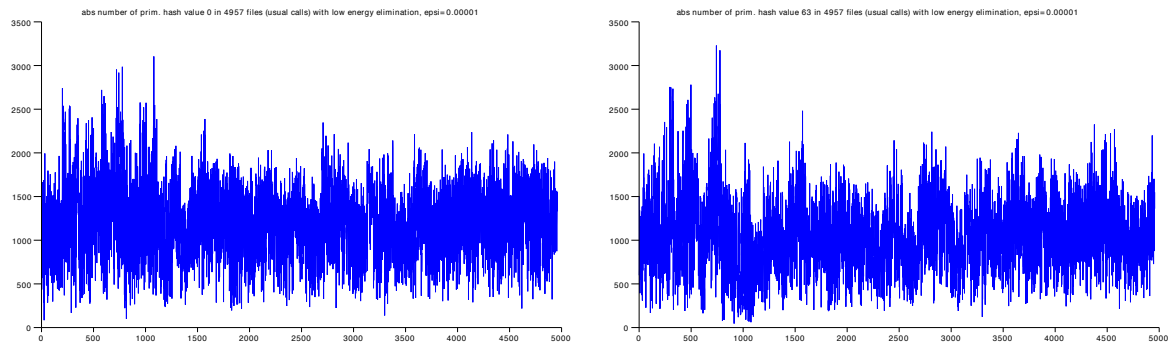

Figure 22: Occurrences of the primary hash values 0 (left) and 63 (right) for all large enough regular calls with elimination of low energy parts with threshold $\varepsilon = 10^{-5}$

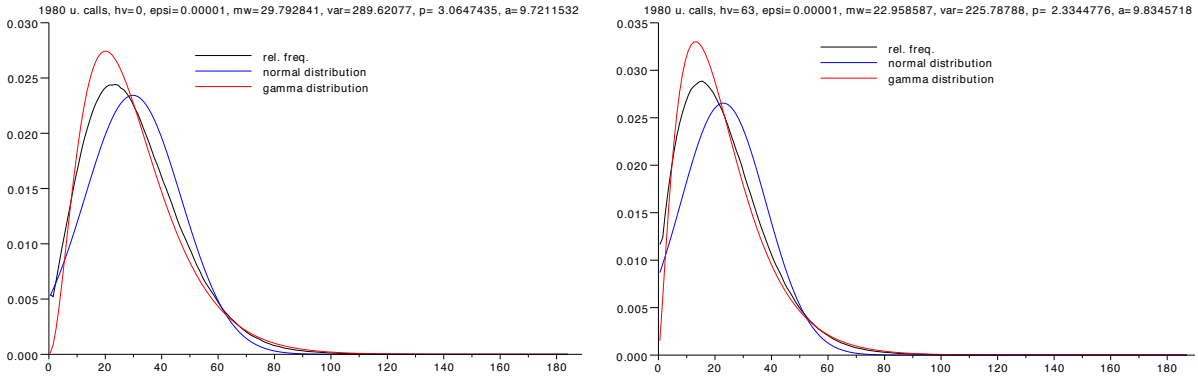

Figure 23: Relative frequency of matches for regular calls for primary hash value 0 (left) and primary hash value 63 (right) with elimination of low energy parts with threshold $\varepsilon = 10^{-5}$

table 7 and 8 (in the appendix, to be compared with table 5 and 6). For 0 as the primary hash value, 5 former "problem pairs" dissappear. The pair "SPIT_07_DimpelM1_n20p" and "SPIT_07_DimpelM1_32kbps" for example is now clearly recognized with 626 matches. But for the primary hash value 63, there are now 14 problem pairs instead of 13 without low energy elimination, one former problem pair is now recognized, but two formerly recognized pairs have now a number of matches below the threshold.

For our special spam files (obtained from regular calls), elimination of low energy parts is disastrous for spam detection as is shown in figure 24. For primary hash value 0 we have 79 undetected spam pairs, for primary hash value 63 we have 170 undetected pairs (for a total number of 180 pairs to be detected). This means, for the primary hash value 63 less than 6 % of the spam pairs are detected. It should be remarked here that the five audio files token for the generation of different versions of "special spam" files have been chosen to produce problem cases using detailed knowledge of our spam detection. The reason for the poor result is visible in figure 25. Those five files have been chosen as the 5 files with minimal occurrences of primary hash value 63 with low energy elimination in a subset of our regular test files (in the set of "special spam" files they have the numbers 1, 10, 19, 28 and 37). In particular, for the file "g003aB7"(file number 10) and most of its subversions (file number 11 to 16), the number of occurrences of the primary hash value 63 is below the threshold, so it is impossible by our method to recognize them as spam.

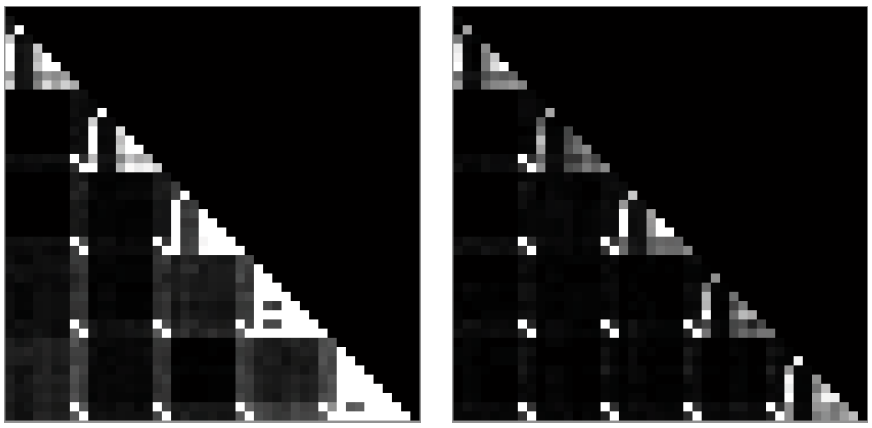

Figure 24: Number of matches for special spam calls for primary hash value 0 (left) and primary hash value 63 (right) with elimination of low energy parts with threshold $= 10^{-5}$

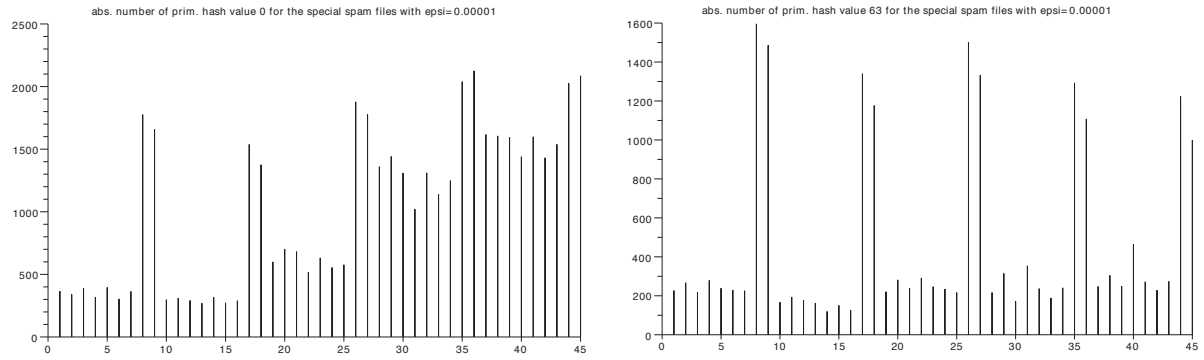

Figure 25: Occurrences of primary hash value 0 (left) and 63 (right) for the special spam calls (generated from 5 regular calls) with elimination of low energy parts with threshold $= 10^{-5}$

Concluding this section it may be stated that elimination of the low energy parts of the signal alone has no significant advantages for the detection of spam by the method presented here, but it decreases the probability of false positive diagnostic. It should however be kept in mind that the secondary hash values are the time indices when some characteristic signal properties arise. And the time positions when silence, some characteristic background noises or other low energy signals are dominating might be a quite valuable criterion which is robust against the modifications considered here, e.g. additional noise.

6 Influence of the offset and offset correction of the audio signals

It should be remarked that the signals of the test corpus contain contributions of very low frequency and even an offset. This is shown in figure 26. It is visible that the offset is robust to most of the changes by which the normal spam versions of the same type differ. It should also be remarked that the offset of some spam versions is much larger than all offsets of the regular calls. An offset in an audio signal is not considered to have much

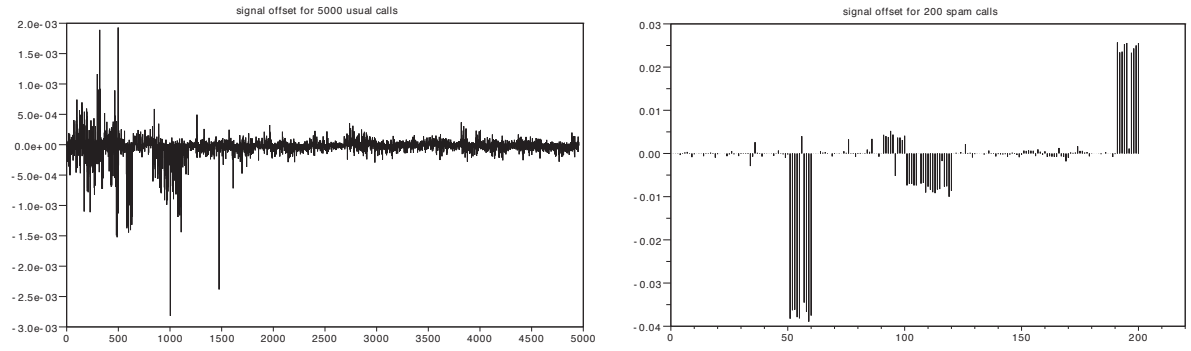

Figure 26: Offset of the signals of the test corpus, at the left for the regular calls, at the right for all versions of the normal spam calls

importance. One may eliminate the offset by replacing the vector $\mathbf{a}$ of the audio samples by the following "offset corrected" vector $\mathbf{a} - \overline{\mathbf{a}}$ where $\overline{\mathbf{a}}$ is the mean of $\mathbf{a}$. The following observations have been made introducing this correction of the offset to zero:

- The exceptional number of matches for the pair of regular calls "g215aB5" and "g218aB6" decreases from 1653 to 1640.
- The number of occurrences of the primary hash values 0 and 63 is changed qualitatively, this number is much more varying from file to file and more files have very few occurrences, compare figure 27 with 10.

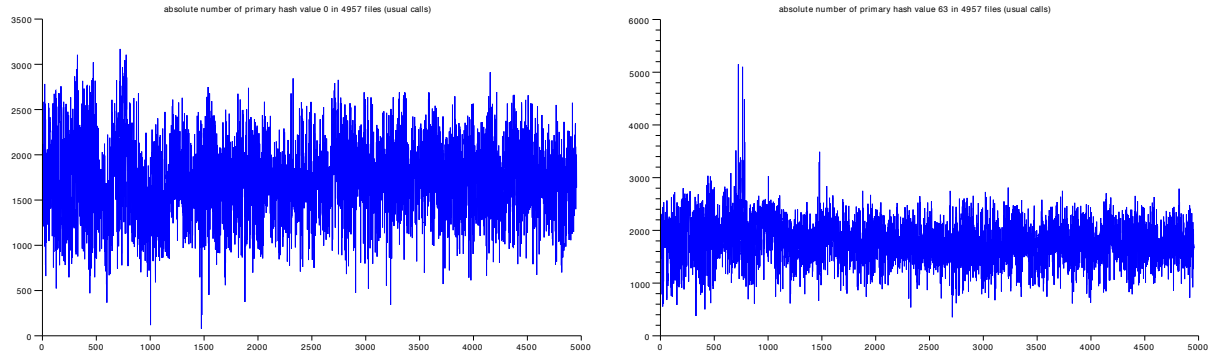

Figure 27: Occurrences of primary hash value 0 (left) and 63 (right) in 4957 files for regular calls with offset correction

- The relative frequency of matches for regular calls in an experiment for more than $1.9 \cdot 10^6$ randomly chosen pairs is changed qualitatively which is visible in figure 28 (comparing it with figure 11); in particular for low values the curve looks quite different. For high values, the frequency stays now below the value of the gamma distribution, at least as far as it is visible.
- The relative frequency of false positive diagnostics is changed for the two primary hash values 0 and 63 in the opposite direction: offset correction causes a decrease for 0 but an increase for 63. This may be seen comparing figure 29 with figure 13. Again the filenames of the pairs in table 4 (in the appendix) are near together in

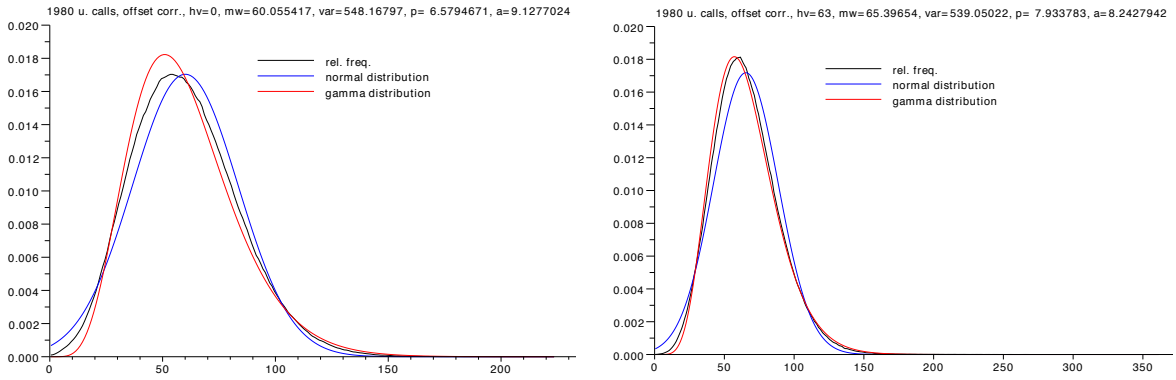

Figure 28: Relative frequency of matches for regular calls with offset correction and primary hash values 0 (left) and 63 (right)

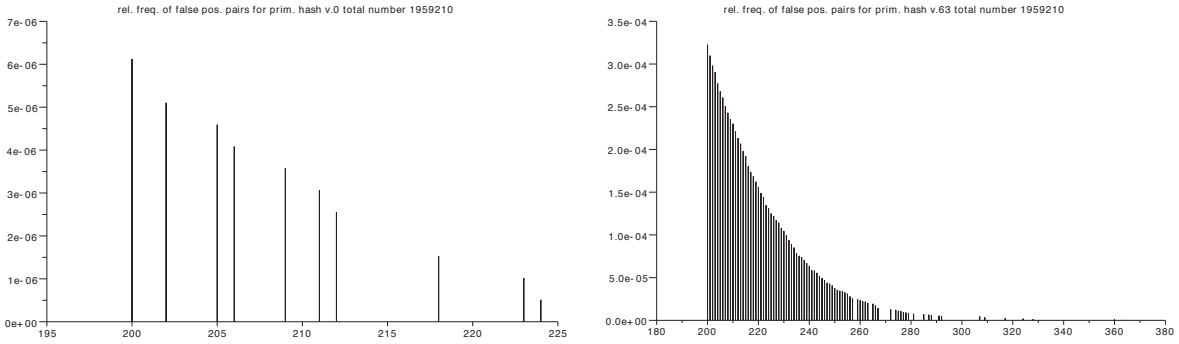

Figure 29: Decrease of the relative frequency of false positive diagnostics with offset correction if the threshold is increased for primary hash value 0 (left) and 63 (right), a total number of more than $1.9 \cdot 10^6$ pairs have been analyzed

the alphabetic list and not randomly distributed. Note however the increase of the decay parameter of the gamma distribution for both primary hash values, i.e. we get a stronger decay for increasing number of matches. In a further experiment (the same number of pairs, again randomly chosen) we got 12 pairs with a number of matches at or above the threshold for the primary hash value 0, but for the primary hash value 63 we got 632 pairs of regular calls which would have been classified as spam, which is significantly worse than without offset correction.

- Detection of normal spam is better. For primary hash value 0, the number of "problem pairs" is reduced from 56 to 12, for the primary hash value 63, the reduction is from 13 to 8, compare table 9 and 10 with 5 and 6

- For our special spam files and primary hash value 0, we get 80 undetected pairs (44 %). This is much worse than without offset correction. For primary hash value 63, we obtain 33 undetected pairs (18 %) which is an improvement. The result is visualized in figure 30.

Concluding this section it may be stated that offset correction alone seems to have some advantages for spam detection, but it increases the probability of false positive diagnostics for primary hash value 63.

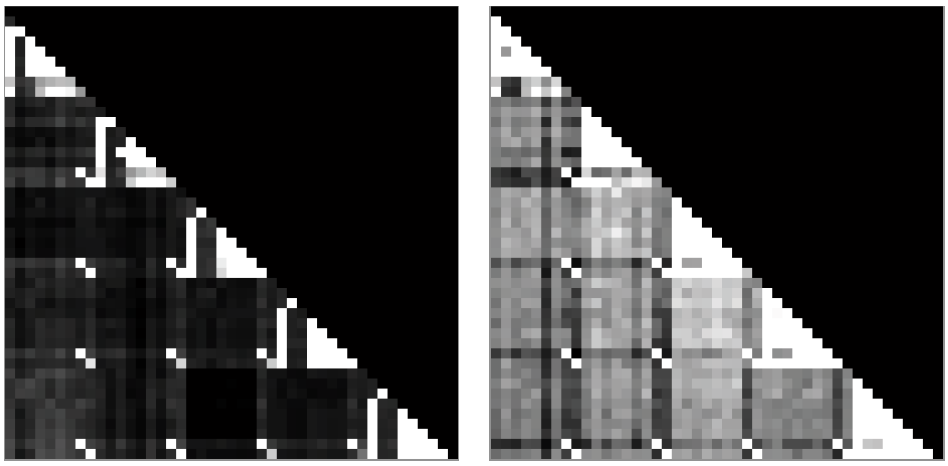

Figure 30: Spam detection for special spam les, o set correction and primary hash values 0 (left) and 63 (right)

7 Combining o set correction and low energy elimination

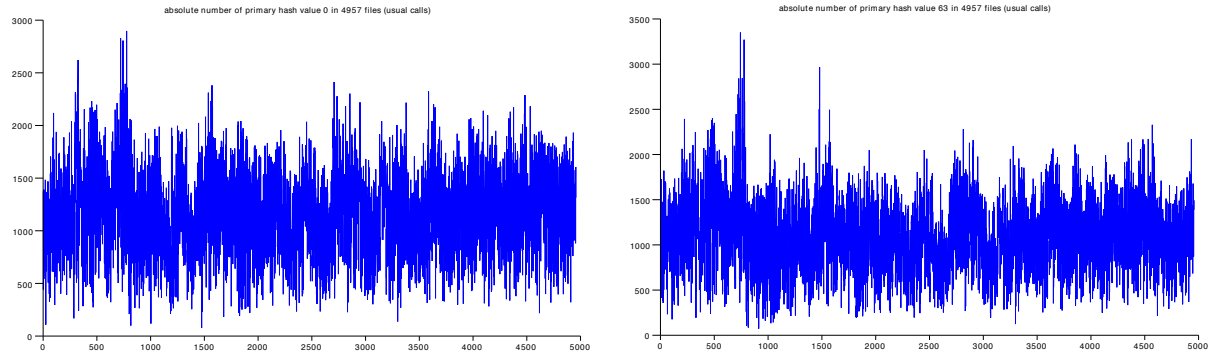

Figure 31: Occurrences of primary hash value 0 (left) and 63 (right) in 4957 les for regular calls with both o set correction and low energy elimination

The o set correction described in the previous section may be combined with the elimination of low energy parts of the signal (see section 5). We chose a threshold of $= 10^{-5}$ for the energy and made the following observations:

Adding o set correction to low energy elimination slightly lowers the occurrences of the primary hash value 0, for the primary hash value 63, there seems to be no signi cant change (see gure 31).

Elimination of low energy parts with o set correction further improves (normal) spam detection only for primary hash value 0 for which the number of undetected spam pairs is further decreased from 12 to 8, for the primary hash value 63 we have the same undetected pairs (as for low energy elimination alone), the number of matches is partially changed, see table 11 and 12 (in the appendix).

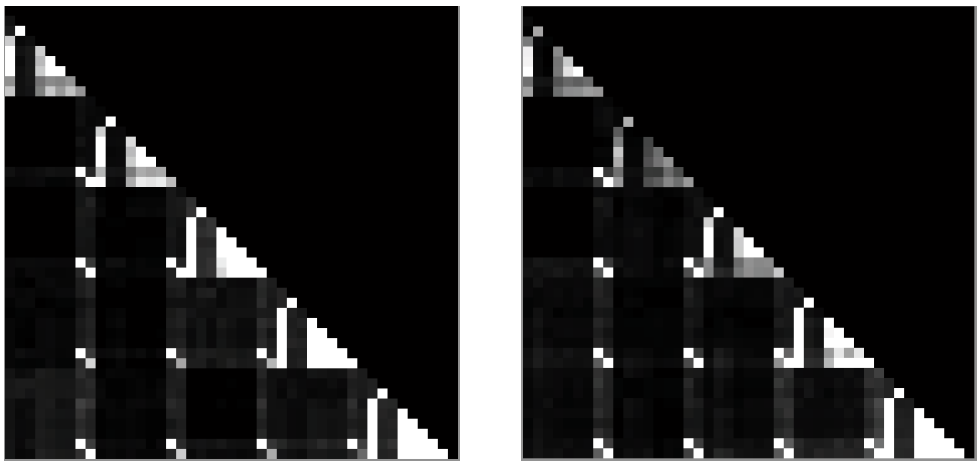

Figure 32: Spam detection for special spam les with both o set correction and low energy elimination for primary hash values 0 (left) and 63 (right)

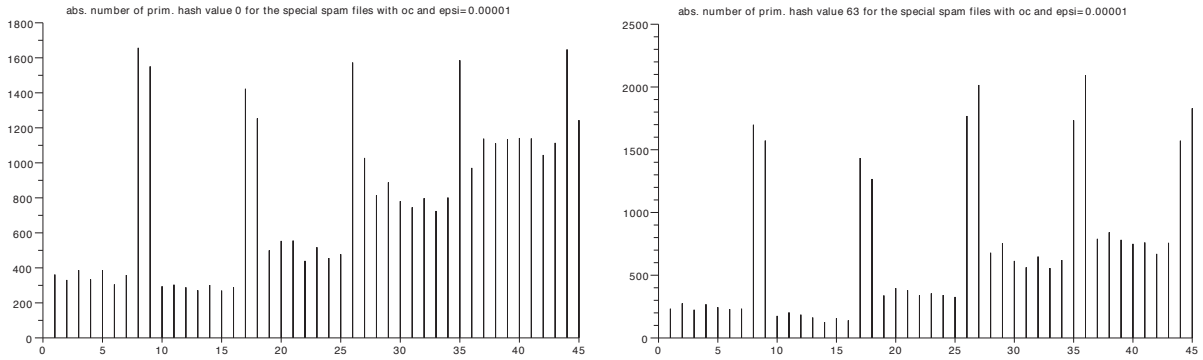

Figure 33: Occurrences of primary hash value 0 (left) and 63 (right) for the special spam calls (generated from 5 regular calls) with both elimination of low energy parts with threshold $= 10^{-5}$ and o set correction

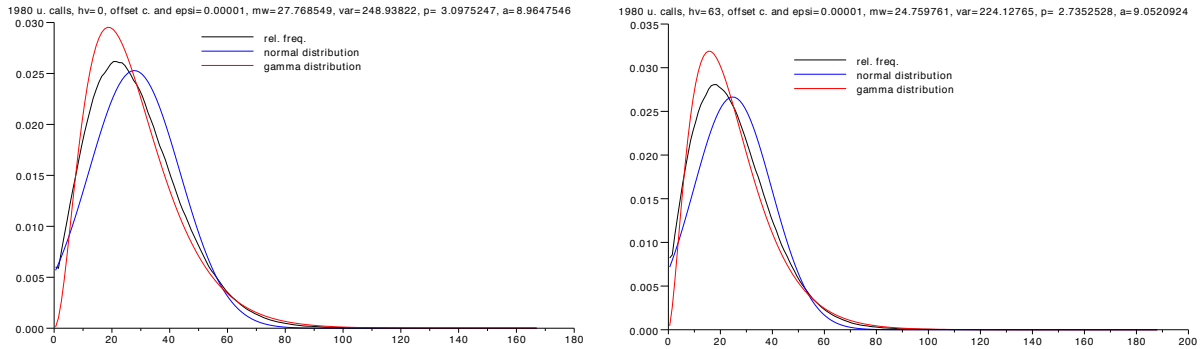

Figure 34: Relative frequency of matches for regular calls with o set correction and low energy elimination for primary hash values 0 (left) and 63 (right)

Detection of the special spam signals is very bad again. For the primary hash value 0, we get 102 undetected pairs (57 %), and for the primary hash value 63, we get 133 undetected pairs (74 %). Note however at the visualization of the number of

matches in figure 32 that our method indicates clearly the similarity of most special spam files of the same type. The problem is the low absolute number of matches compared to those for pairs of most of the regular calls which is caused by the low number of the corresponding primary hash values shown in figure 33.

- In an experiment with more than $1.9 \cdot 10^6$ pairs and threshold 200, for primary hash value 0 there has been no pairs detected with false positive diagnostic, and for primary hash value 63 there was just one such pair, in a further experiment with other randomly chosen pairs there was no such such pair for both primary hash values. The relative frequency of the number of matches for those experiments is quite similar to the case of low energy elimination without offset correction, it is shown in figure 34.

Those experiments indicate that it seems to be most advantageous for our method to include both offset correction and low energy elimination.

8 Conclusion

The use of the zero crossings of part of the redundant wavelet coefficients turns out to offer quite remarkable possibilities to obtain characteristic values for audio signals which are robust against modifications as the addition of noise. Those characteristic values may be applied to detect the similarity of audio signals and this may be used to detect telephone spam without analyzing the semantic content by speech recognition. Instead of the index position of the zero crossings, here the sign of the coefficients at each index position is used. It turns out that the time indices when all the coefficients considered are positive or zero — which corresponds to our primary hash value 63 — or when they all are negative — which corresponds to our primary hash value 0 — are particularly useful. In practice, it will possibly have many advantages to use them both in combination. This might avoid the difficulty to recognize similar signals when one of them has very few samples where the primary hash value 0 or 63 alone occurs.

The proposed method may be improved by subtracting the mean of the original audio signal to get zero offset and by elimination of the coefficients representing low energy parts of the signal. Table 2 resumes the essential results for the original method and those additional changes. As indicator of the success we used the following calculated values

- the decay rate parameter a resulting from the assumption that the number of matches of regular calls is gamma distributed (with probability density function $f(x) = \frac{a^p}{\Gamma(p)} x^{p-1} e^{-ax}$, a larger value of a should make false positive results less probable which is desirable)

- the number n_u of undetected normal spam pairs

- the number n_f of false positive diagnostics for about $1.9 \cdot 10^6$ randomly chosen pairs of regular calls

- the number s_u of undetected special spam pairs

Those values have been calculated for the primary hash values $h_p = 0$ and $h_p = 63$ for the following variants of our method

① without low energy elimination and without offset correction (see section 4)

② with low energy elimination using a threshold of $\varepsilon = 10^{-5}$ but without offset correction (see section 5)

③ without low energy elimination but with offset correction (see section 6)

④ with both low energy elimination using a threshold of $\varepsilon = 10^{-5}$ and offset correction (see section 7)

Note that for the calculation of the number of false positive diagnostics n_f indicated in table 2 a special experiment with the same randomly chosen pairs (about $1.9 \cdot 10^6$ pairs) for all 8 rows has been performed for better comparison. Slightly different results with other randomly chosen pairs have been mentioned in previous sections.

variant	section	h_p	a	n_u	n_f	s_u
①	4	0	3.62	56	33	36
①	4	63	4.78	13	248	91
②	5	0	9.68	53	0	79
②	5	63	9.72	14	0	170
③	6	0	9.13	12	12	80
③	6	63	7.93	8	632	33
④	7	0	8.96	8	0	102
④	7	63	9.05	8	0	133

h_p	primary hash value
a	decay parameter for the number of matches for regular calls
n_u	number of undetected normal spam pairs
n_f	number of false spam diagnostics
s_u	number of undetected special spam pairs

Table 2: Résumé of essential results for variants of the method described here; "section" refers to the section, where the variant is described

The results indicate clearly that combining low energy elimination and offset correction is the most advantageous variant with the caveat that their might be a possibility to prevent the detection using spam signals with specially adapted properties if the detail and the chosen variant of the method is known — as is illustrated by our special spam signals.

For the test corpus investigated here — without the special spam signals — the results presented show that we may detect nearly all pairs of different versions of the same spam message. The possibility of false positive diagnostics is quite low. It is remarkable that the method presented here makes even the similarity visible of most of the different spam signals of the test corpus to which the same calculated noise has been added.

Our results indicate that the use of the redundant wavelet transform may have advantages compared to the usual Fourier methods and also to the use of the wavelet transform with subsampling. It would be of interest to analyze those advantages in a detailed quantitative study. From the point of view of future practical use, the main problem will be to implement a fast algorithm to calculate the intersection of two already ordered sets of non negative integers.

The detection of similarity without analyzing the content of the message (which is wished in order to respect the privacy of telephone calls) includes always the risk that this identification is due to some particularity which is common to otherwise completely different signals or which is different for an otherwise similar content. This means that there is a risk of false positive diagnostics which is not due to random effects — or a risk not to detect similar signals.

The parameters and the variants of the method presented here have been chosen to illustrate the qualitative properties of our method. Before drawing more detailed conclusions of the corresponding results, it should be noted that first of all, the test corpus used here should be critically investigated as it presents many particularities which deserve further attention. Some parameters for the regular calls have values that are not at all random as they are qualitatively different for the first 1 000 files compared to the rest (see for instance figure 26 and 27), the difference may not be explained by random effects. Some care is therefore needed interpreting our results based on randomly chosen subsets of those regular calls. Similarly, parameter values of some types of the normal spam signals are qualitatively different from those of the other types which can be seen in figure 26. And those parameter values influence essentially the success of our method.

If a method for spam detection is known in detail, it is in many cases possible to generate spam signals which are quite difficult to be detected by this method. This has to be kept in mind analyzing the relatively poor results concerning our special spam files. It has been mentioned already that they have been generated from regular calls with very few occurrences of the primary hash value 63 (with low energy elimination), therefore making our method particularly unsuccessfull.

Acknowledgements

The author is greatly indebted to Heiko Knospe, Julian Strobl and Gary Grutzek for many fruitful discussions and to Gary Grutzek for providing the test corpus of "special" spam files used here. Thank is also given to the Institut Galilée at Université Paris 13 and Polytech' Marseille, Filière Microélectronique et Télécommunications, for their hospitability; part of the reported work has been carried out there.

References

[1] N. Chen, W. Wan, and H.-D. Xiao. Robust audio hashing based on discrete-wavelet-transform and non-negative matrix factorisation. *IET Communications*, 4(14):1722 – 1731, September 2010.

[2] Gary Grutzek, Julian Strobl, Bernhard Mainka, Frank Kurth, Christoph Pörschmann, and Heiko Knospe. Perceptual hashing for the identification of telephone speech. Technical report, Institute of Communications Engineering, Cologne University of Applied Sciences, 50679 Cologne, Germany, 2012.

[3] Jianguo JIANG, Kaige MA, Mingxing WEN, and Yongqing LIU. An audio fingerprint algorithm based on statistical characteristics of db4 wavelet. *Journal of Information and Computational Science*, 8(14):3027–3034, 2011.

[4] S. Mallat. Zero-crossings of a wavelet transform. *IEEE Trans. Inform. Theory*, 37(4):1019–1033, 1991.

[5] Alexander Stoffel. Remarks on the unsubsampled wavelet transform and the lifting scheme. *Signal Processing*, 69(2):177–182, 1998.

A Appendix

filename 1	filename 2	n_m
g241aB3	g013aA7	208
g215aB2	g014aA4	202
g239aB3_2	g014aA5	216
g215aB2	g014aA8	210
g215aB2	g016aA2	216
g232aB9_2	g016aA2	210
g215aB2	g016aA9	207
g215aB1	g203aB1	213
g239aB3_2	g203aB7	202
g231aB2	g213aB7	213
g215aB2	g215aB1	201
g216aB5	g215aB1	203
g234aB5_2	g215aB1	209
g241aB3	g215aB1	227
g216aB5	g215aB2	227
g231aB2	g215aB2	204
g232aB8	g215aB2	208
g232aB9_2	g215aB2	214
g234aB2	g215aB2	206
g234aB5_2	g215aB2	204
g239aB3_2	g215aB2	228
g241aB3	g215aB2	205
g234aB5_2	g216aB5	218
g239aB3_2	g216aB5	226
g241aB3	g216aB5	204
g234aB2	g218aB5	205
g234aB5_2	g232aB9_2	228
g239aB3_2	g232aB9_2	214
g447aB4_2	g232aB9_2	205
g239aB3_2	g234aB2	200
g239aB3_2	g234aB5_2	202
g239aB3_2	g237aB3_2	200
g241aB3	g239aB3_2	209

Table 3: Pairs of regular calls with false positive diagnostics without offset correction for primary hash value 0, note that in the alphabetic list, the filenames go from g001aA1_2 to g931aB11; n_m denotes the number of matches

filename 1	filename 2	n_m
g018aB4	g012aB7	200
g215aB2	g018aB1	205
g046aB8	g018aB4	218
g215aB1	g018aB4	212
g215aB2	g018aB4	200
g227aB8	g018aB4	223
g215aB2	g018aB5	224
g218aB5	g018aB5	211
g046aB8	g046aB2	206
g215aB2	g046aB8	209
g216aB5	g046aB8	212
g216aB5	g215aB2	202

Table 4: Pairs of regular calls with false positive diagnostics with offset correction for primary hash value 0, note that in the alphabetic list, the filenames go from g001aA1_2 to g931aB11; n_m denotes the number of matches

filename 1	filename 2	number of matches
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_32kbps	121
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_96kbps	119
SPIT_07_DimpelM1_n20p	SPIT_07_DimpelM1_32kbps	111
SPIT_11_Hermans1_n20p	SPIT_11_Hermans1_32kbps	180
SPIT_11_Hermans1_n20p	SPIT_11_Hermans1_96kbps	172
SPIT_12_Hermans2_GSM	SPIT_12_Hermans2_32kbps	74
SPIT_12_Hermans2_n20p	SPIT_12_Hermans2_32kbps	156
SPIT_12_Hermans2_n20p	SPIT_12_Hermans2_96kbps	149
SPIT_18_MrSmith1_GSM	SPIT_18_MrSmith1_32kbps	66
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_32kbps	73
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_96kbps	69
all 45 signal pairs of type "SPIT_20_unbek" with 90 matches or less		

Table 5: Spam detection with shift correction: undetected normal spam pairs using primary hash value 0

filename 1	filename 2	number of matches
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_32kbps	65
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_96kbps	68
SPIT_04_BMW_GSM	SPIT_04_BMW_32kbps	154
SPIT_04_BMW_GSM	SPIT_04_BMW_96kbps	155
SPIT_07_DimpelM1_GSM	SPIT_07_DimpelM1_32kbps	113
SPIT_07_DimpelM1_n20p	SPIT_07_DimpelM1_32kbps	72
SPIT_07_DimpelM1_n20p	SPIT_07_DimpelM1_96kbps	72
SPIT_18_MrSmith1_GSM	SPIT_18_MrSmith1_32kbps	81
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_32kbps	82
SPIT_18_MrSmith1_GSM	SPIT_18_MrSmith1_96kbps	124
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_96kbps	91
SPIT_20_unbek_GSM	SPIT_20_unbek_32kbps	176
SPIT_20_unbek_GSM	SPIT_20_unbek_96kbps	177

Table 6: Spam detection with shift correction: undetected normal spam pairs using primary hash value 63

filename 1	filename 2	number of matches
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_32kbps	121
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_96kbps	117
SPIT_04_BMW_GSM	SPIT_04_BMW_32kbps	57
SPIT_11_Hermans1_n20p	SPIT_11_Hermans1_32kbps	167
SPIT_11_Hermans1_n20p	SPIT_11_Hermans1_96kbps	144
SPIT_17_Rademchr2_n20p	SPIT_17_Rademchr2_32kbps	134
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_32kbps	73
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_96kbps	67
all 45 signal pairs of type "SPIT_20_unbek" with 90 matches or less		

Table 7: Spam detection with shift correction: unrecognized signal pairs with low energy elimination with threshold $\varepsilon = 10^{-5}$ using primary hash value 0

filename 1	filename 2	number of matches
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_32kbps	65
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_96kbps	68
SPIT_04_BMW_GSM	SPIT_04_BMW_32kbps	147
SPIT_04_BMW_GSM	SPIT_04_BMW_96kbps	150
SPIT_04_BMW_n20p	SPIT_04_BMW_96kbps	101
SPIT_07_DimpelM1_n20p	SPIT_07_DimpelM1_32kbps	66
SPIT_07_DimpelM1_n20p	SPIT_07_DimpelM1_96kbps	67
SPIT_11_Hermans1_n20p	SPIT_11_Hermans1_96kbps	36
SPIT_18_MrSmith1_GSM	SPIT_18_MrSmith1_32kbps	80
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_32kbps	82
SPIT_18_MrSmith1_GSM	SPIT_18_MrSmith1_96kbps	124
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_96kbps	91
SPIT_20_unbek_GSM	SPIT_20_unbek_32kbps	167
SPIT_20_unbek_GSM	SPIT_20_unbek_96kbps	167

Table 8: Spam detection with shift correction: unrecognized signal pairs with low energy elimination with threshold $\varepsilon = 10^{-5}$ using primary hash value 63

filename 1	filename 2	number of matches
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_32kbps	113
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_96kbps	114
SPIT_04_BMW_GSM	SPIT_04_BMW_32kbps	79
SPIT_04_BMW_GSM	SPIT_04_BMW_96kbps	102
SPIT_04_BMW_n20p	SPIT_04_BMW_96kbps	117
SPIT_07_DimpelM1_GSM	SPIT_07_DimpelM1_32kbps	132
SPIT_07_DimpelM1_n20p	SPIT_07_DimpelM1_32kbps	113
SPIT_07_DimpelM1_GSM	SPIT_07_DimpelM1_96kbps	116
SPIT_12_Hermans2_GSM	SPIT_12_Hermans2_32kbps	37
SPIT_12_Hermans2_GSM	SPIT_12_Hermans2_96kbps	41
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_32kbps	114
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_96kbps	126

Table 9: Spam detection with shift and offset correction : unrecognized signal pairs using primary hash value 0

filename 1	filename 2	number of matches
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_32kbps	76
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_96kbps	79
SPIT_04_BMW_g726_16	SPIT_04_BMW_32kbps	166
SPIT_04_BMW_GSM	SPIT_04_BMW_96kbps	95
SPIT_07_DimpelM1_n20p	SPIT_07_DimpelM1_32kbps	75
SPIT_07_DimpelM1_n20p	SPIT_07_DimpelM1_96kbps	75
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_32kbps	82
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_96kbps	85

Table 10: Spam detection with shift and offset correction : unrecognized signal pairs using primary hash value 63

filename 1	filename 2	number of matches
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_32kbps	100
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_96kbps	110
SPIT_04_BMW_GSM	SPIT_04_BMW_32kbps	79
SPIT_04_BMW_GSM	SPIT_04_BMW_96kbps	97
SPIT_12_Hermans2_GSM	SPIT_12_Hermans2_32kbps	37
SPIT_12_Hermans2_GSM	SPIT_12_Hermans2_96kbps	41
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_32kbps	115
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_96kbps	123

Table 11: Spam detection with shift and offset correction and elimination of low energy parts with threshold $\varepsilon = 10^{-5}$: unrecognized signal pairs using primary hash value 0

filename 1	filename 2	number of matches
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_32kbps	76
SPIT_02_ABeier2_n20p	SPIT_02_ABeier2_96kbps	79
SPIT_04_BMW_g726_16	SPIT_04_BMW_32kbps	162
SPIT_04_BMW_GSM	SPIT_04_BMW_96kbps	95
SPIT_07_DimpelM1_n20p	SPIT_07_DimpelM1_32kbps	71
SPIT_07_DimpelM1_n20p	SPIT_07_DimpelM1_96kbps	71
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_32kbps	82
SPIT_18_MrSmith1_n20p	SPIT_18_MrSmith1_96kbps	85

Table 12: Spam detection with shift and offset correction and elimination of low energy parts with threshold $\varepsilon = 10^{-5}$: unrecognized signal pairs using primary hash value 63