An analysis of one of the SDMI candidates

Julien Boeuf[1] and Julien P. Stern[2][3]

[1] École Nationale Supérieure des Telecommunications,
46, rue Barrault, F-75634 Paris Cedex 13, France

[2] Laboratoire de Recherche en Informatique,
Université de Paris-Sud,
Batiment 490, F-91405 Orsay Cedex, France
stern@lri.fr

[3] UCL Crypto Group,
Batiment Maxwell, Place du levant, 3
B-1348 Louvain-la-Neuve, Belgique
stern@dice.ucl.ac.be

Abstract. Watermarking technologies have received a considerable attention from both the academic and the industrial worlds lately. The first very large scale deployment of a intellectual property protection system relying on watermarking might be led by the Secure Digital Music Initiative consortium (SDMI). On September 6th 2000, the Executive Director of the SDMI group published an "open letter to the digital community" [otSg], where he invited hackers to try to break the technologies developed by several members of the SDMI group. In this paper, we present the detailed analysis of one of the proposed watermarking schemes, and explain how to defeat it.

1 Introduction

The tremendous growth of the Internet nowadays makes the copy and distribution of any kind of digital data easy. This raises the problem of intellectual property protection on these data, which are sometimes illegally copied material. The problem is especially present for digital music, due to its attractiveness and to the availability of efficient compression algorithms, such as MP3. This phenomenon is aggravated by the development of peer-to-peer sharing systems, where anyone can access files stored on the computers of any other participating user.

In order to fight against piracy, a number of companies have gathered to form the Secure Digital Music Initiative (SDMI), whose goal is to develop technologies to protect the playing, storing and distribution of digital music.

Recently, the SDMI has selected a number of candidate technologies that could be part of the final deployed system, and has launched a public challenge [otSg] in order to test the robustness of these candidates.

In this paper, we present the analysis of one of these candidate technologies and explain how to defeat it. As we will see, this technology is based on the spread-spectrum technique, often used in watermarking. We also envision a more general setting than the one used in the challenge, which allows us to pinpoint an intrinsic weakness to spread-spectrum based scheme: collusion attacks. This weakness forces a large number of spread-spectrum based scheme, to rely, not only on the secrecy of their private informations but also on the secrecy of their design to be secure.

We present the general security framework of the SDMI in section 3, and detail the kind of attacks than can be attempted against a watermarking scheme in section 4. In section 5, we explain how we analyzed the proposed scheme and we present the attack in section 6.

2 Related Work

There has been a considerable amount of work on the subject of watermarking in the last few years. The most promising systems today are based on spread spectrum techniques. Such techniques were introduced in watermarking by [CKLS96,HG96], and have been largely improved in many different publications. We refer the reader to [Auc98,Pfi99] for an overview.

A summary of attacks aimed at defeating watermarking schemes in general can be found in [PAK98], and a recent survey on information hiding techniques in [PAK99].

Regarding the SDMI challenge, we are aware of the effort of other researchers, who claim to have defeated all the SDMI schemes [CMW$^+$]. Their technical report was to available at the time of this writing.

3 A brief overview of the SDMI framework

The scenario is the following: there will be SDMI compliant devices which may be of different kinds (HiFi, portable players, car players, etc). In order to play a song on such a device, it needs to pass the gate of the *secure world.* To make a song enter the secure world, in has to be on an LP. The LP is then checked to insure that is it a "legal" one.

The two main requirements are the following:

- All legacy LPs must pass this gate.
- All new LPs that have been legally bought must also pass this gate.

The goal of the SDMI is to prevent the following: Bob buys an LP, rips the tracks to his computer, compress them, sends them to Alice. Alice burns them on an LP and imports them into the *secure world.*

What their algorithm does not prevent (although it's illegal) is the following: Bob buys an LP, burns a copy, gives the copy to Alice. Alice imports the songs into the *secure world.* As a matter of fact, it seems impossible to make the difference between an original LP and a perfect copy of it. Consequently, if Bob transmits an ISO image of the LP over the net to Alice, she should be able to burn it and import it into the *secure world.* However, such an image is very big, and this procedure is time consuming and may be costly.

To put it another way, it should be impossible to import an LP into the *secure world* if it has been modified in any way (notably if it has been compressed). Checking for the integrity of a document can be done using standard cryptographic techniques, such as MAC, or even signatures. Therefore, one can wonder at first why watermarking is needed? The problem is that legacy LPs do not include any kind of verification information but should not be rejected. Consequently, it is necessary to be able to distinguish legacy and new LPs. There is where watermarking technologies will be used.

3.1 The gatekeeper

Let us now detail how the entrance to the secure world is checked.

As one may easily figure, two algorithms are going to be involved both in the creation and the detection processes.

- An identification technology (Technologies named D and E during the hackSDMI challenge)
- A watermarking technology (Technologies named A,B,C and F during the hackSDMI challenge)

When an LP is created, the songs on the LP are watermarked using the watermarking technology. Then, the LP is "signed" (we do not know whether this is actually a cryptographic signature) using the identification technology.

The watermarking technology is simply here to enable the gatekeeper distinguish legacy content and new content. If a mark can be found, the content is deemed new.

The identification technology is here to prevent modifications of the LP, notably compression.

When an LP is trying to enter the secure world, the following checks are made:

- Is it marked?
- Is there a signature and is it valid?

Therefore, we have several cases, presented in figure 1.

	Marked	Non Marked
Signed and invalid	Reject	Reject?
Unsigned	Reject	Accept
Signed and valid	Accept	Accept?

Fig. 1. Decision to accept or reject the entrance of a song in the *secure world* based on the gatekeeper tests

The results of the first column are very clear. If the mark is found, the LP is considered new, therefore it should be correctly signed. If it is not, it should be rejected.

The second column is not as clear. If it is unmarked and unsigned, it is supposed to be a legacy LP so it should be accepted. If it is unmarked BUT signed, it means something strange has happened. We are not sure how the test behaves in these cases and the above table only represent our guess.

3.2 How to attack this system?

There are two ways to attack the system: one can try to defeat either the identification technology or the watermarking one.

Breaking the identification technology is rather unlikely to succeed, because digital signatures are safe unless a design error is made.

The second attempt is to remove the marks so that the detector believes that the LP is a legacy one. This is the method that we chose to attempt.

3.3 The HackSDMI challenge

On September 6th 2000, the Executive Director of the SDMI group published an "open letter to the digital community" [otSg], where he invited hackers to try to break the technologies developed by several members of the SDMI group.

The challenge was divided into six parts, each of them consisting of a different technology. There was a cash prize of $10000 for the breakage of each technology provided you agreed to be bound by a Non Disclosure Agreement.

Four of the challenges (named A,B,C and F) were watermarking technologies. The two others were probably essentially digital signature technologies.

We were primarily interested in the watermarking technologies. The information given for the other two was much less than what could have been expected in a fair challenge.

Each watermarking challenge included three songs, two copies (one marked and one unmarked) of a first song, and a marked version of a second. The goal of the challenge was to remove the mark from this second song. The success of the challenge was assessed by an "Oracle" which could receive songs through a web interface and indicate whether the attack was successful. The inner working of the Oracle were not well specified. From our experiments, it seemed that it was both checking for the presence of the mark, and also testing the "quality" of the music in some automated way.

The material provided is naturally largely below the usual settings of a cryptographic analysis. One could have expected to have a larger sample of songs to ease statistical analysis, a faster access to the Oracle or even the details of the marking algorithms.

In spite of this, we were able to almost fully analyze one of the schemes, the watermarking technology F, which we present and attack below.

4 Attack definitions

Before explaining how we analyzed the technology, let us informally detail the different kinds of attacks that can be attempted against a watermarking scheme.

Let us call A the original version of a song, and AM the marked version. Obviously, recovering A from AM, or something infinitely close to A will remove the mark. However, it is not always necessary to perform such a hard task. It might sometimes be enough to produce a new song C which is simply reasonably close to A, but which has the property that the mark cannot be detected in it anymore.

A simple example of this second type of attack would be to have musicians re-record the song. Clearly, the mark will not be present. The problem is to be able to produce something close enough to this original.

Actually, it is not even always necessary to "remove" the mark, it may be that the mark is still present, in some sense, but that the structure of the song has changed in such a way that the detector cannot find it anymore. This kind of attacks, that are sometimes called desynchronisation attacks, are known to be very efficient against images. While some papers have tried to explicitly fight them [PP99,JDJ99], geometric transformations on marked images make detection extremely hard if not impossible.

Let us now define the attacks that can be applied to defeat the proposed schemes. Our definitions are informal ones and should be seen from the practical point of view of an attacker.

Random attacks To perform a random attack one does not need to understand anything on how the marking algorithm works. The idea is simply to apply some kind of transformation to the song, and hope the mark cannot be detected anymore. This attack can be applied very easily but is rarely successful.

If one can get fast answers from an oracle (which was not the case in the contest), then maybe eventually, one will obtain a song where the mark cannot be detected anymore. It is in fact possible to work with a kind of dichotomy: one can very strongly degrade a song to obtain an unmarked version and try to find a song in the middle which would have the properties of not carrying the mark and yielding an acceptable quality.

Clearly, one might need a huge amount of trials before succeeding, and the quality of the final result is not guaranteed.

Directed attack To mount this attack, one needs a partial understanding of the watermarking scheme. The idea is to apply a transformation of a similar kind as the one used for marking. For example, if it is known that the marking process is only modifying the phase of the signal, one can try to apply all-pass filters or similar transformations. This is not in theory extremely different from the previous attack, but in practice, the transformations performed are less likely to degrade the song (because they change only part of it), and are also more likely to remove the mark (because they modify the same parameters).

Naturally, the better one understands the scheme, the highest are his chances of success. The problems with this attack are that (a) one cannot be sure the attack will work against every song, (b) one cannot be sure the final audio quality will be good enough for every song. Consequently, this attack cannot really be automated.

Surgical attack The surgical attack is the ultimate version of the directed attack. It requires an almost complete understanding of the inner workings of the marking scheme. This attack represents the case where one is actually able to recover the original song, by "surgically" removing only the part representing the mark from the marked version. The main advantage of finding such an attack is that all the problems related to the quality of the songs are gone. This attack can consequently be automated. Hence, with a surgical attack, one could code a filter that would automatically remove the mark of any song downloaded to his computer, thus defeating the whole purpose of the scheme.

We were able to mount random or directed attacks against all the schemes proposed by the SDMI, based on non-standard compression and non-linear time modifications. We were also able to perform an almost surgical attack on one of the scheme, which is the one we are going to present now.

5 Analysis

5.1 The challenge material

For each watermarking technology, the challenge included three songs, encoded using the `wav` format. These songs were two minute long, and were sampled at the sampling rate of $Fs = 44100Hz$.

- The first song was an original, never released song. We will denote it by A.
- The second song was the marked version of the first one. We will denote it by AM.
- The last song was a marked, never released song. We will denote it by BM.

The aim of the challenge was to produce a new song C, that should

1. be of a sufficient quality (i.e. better than or equivalent to an MP3 encoding at 64 kbit/s.)
2. fail the detection test (i.e. the detector corresponding to the marking algorithm should not be able to detect the mark.)

Obviously, recovering the original clean song corresponding to BM solves the problem, but as we have seen previously, doing this is much more than is actually required to win the challenge.

5.2 Understanding the marking algorithm

The most natural step to perform with these data is to analyze the difference D between AM and A, that is, the quantity which is actually added to the unmarked version.

We performed an autocorrelation on D, which is shown on figure 2. The regularly spaced peaks indicate that the signal is periodic: we measured this period P, and obtained:

$$P = 1470 \text{ samples } = \frac{1}{30} \text{ sec} \tag{1}$$

Then, we compared two successive periods by making their ratios. The graph of the ratio was a stair function, with 10 different stairs. Figure 3 shows this graph for a specific couple of periods.

The stair structure led us to understand that the same pattern is repeated every 1470 samples but is multiplied by a different factor every 147 samples. Let us denote by w this original pattern.

What we know so far, is that in order to compute the $i - th$ chunk (of 1470 samples) of the final mark that is going to be added to the original song, one has to compute:

$$finalmark_i = \begin{bmatrix} \alpha(s,w,i,1) \\ \alpha(s,w,i,2) \\ \vdots \\ \alpha(s,w,i,10) \end{bmatrix} w \tag{2}$$

where α is a (possibly probabilistic) function depending on the original song s, the original pattern w, the index of the computed chunk i, and the subdivision in this chunk.

We understood that α was essentially the norm of the corresponding 147 sample long chunk:

$$\alpha(s,w,i,j) = \beta(s,w,i,j)||s_i[j]||$$

It is quite natural for α to be proportional to this norm. Doing this allows to hide more information when the signal is stronger. Unfortunately, we were not able to exactly figure out the β function. β probably takes into account the fact that the final result must be between -1 and 1, and also perhaps a psychoacoustic model. We also observed that β seems to be the product of a slowly varying function, and of a constant which changes every second. However, we were not able to use these observations to improve our attack.

5.3 The algorithms

We now present how we think the marking and the detection algorithms work. It should be underlined that these are simply suppositions derived from a very limited amount of material. However, these suppositions seem to fit rather well on the three songs that were provided in the challenge.

Algorithm 1 Marking algorithm: inputs: $w \in [-1,1]^{1470}$, $s \in [-1,1]^m$

```
Output and skip $start$ samples from the original song
while The song is not over do
  $s \leftarrow$ the next 1470 samples of the song
  for $j = 1$ to 10 do
    $s[j] \leftarrow s[j] + \beta||s[j]||w[j]$
  end for
  Output $s$
end while
```

Algorithm 2 Detection algorithm inputs: $w \in [-1,1]^{1470}$, $s' \in [-1,1]^m, p, \delta$

```
Skip $start$ samples (possibly resynchronize by correlation)
while The song is not over do
  $sum \leftarrow 0$
  Get the next $p$ chunks of 1470 samples
  for Each of these chunks do
    $s \leftarrow$ the next 1470 chunk
    for $j = 1$ to 10 do
      $s[j] \leftarrow s[j]/\beta||s[j]||w[j]$
    end for
    $sum \leftarrow sum + s$
  end for
  $Q = sum.w$
  if $Q > \delta$ then
    Outputs “mark found”
  end if
end while
```

Let us now briefly explain why we believe the detection algorithm works this way. The embedded mark is very small. It is actually a *noise* compared to the signal of the song. The standard technique to detect a noise embedded in a signal is *correlation*. However, one needs to correlate on a long enough chunk so that the noise correlation is much larger than the correlation of the signal and the noise.

Consequently, correlating on 1470 samples is not enough to reveal the presence of the mark. This is why we are actually correlating on the average of p chunks of 1470 samples.

We have tested the detection algorithm with two different sizes of p, $p = 30$ (one correlation per second) and $p = 450$ (one correlation every 15 seconds, the maximum detection time required by the original SDMI call for proposals). The results are given in section 6.

6 Attacking the algorithm

6.1 Breaking the challenge

To defeat this watermarking scheme, all we had to do was to recreate the mark that was inserted and to substract it.

The first step was to recover the mark. This was done by renormalizing $D = AM - A$ on periods of 147 samples and averaging the result on periods of 1470 samples.

Then, we could "unwatermark" the second song BM by remultiplying the extracted mark by the corresponding norm in BM and then performing a simple substraction of BM and the newly created mark.

Our results are illustrated by the figures in the appendix. They represent the outputs of *our* detection algorithm, for the first forty seconds of the songs. The x-axis always represents seconds. Correlations are made on periods of either one second (figures 4, 5) or of fifteen seconds (figure 6).

It should be noted that the knowledge of β allows the real detector to perform better than ours, and that the results of the real detector may vary from ours. It is also possible that some elements, like, for example, the inner structure of the mark, allows the construction of a more accurate detector. However, the results of our detector are similar for the two marked songs on the one hand, and for all unmarked song and our newly created one on the other. This lead us to think that our technique allows to remove a proportion of the initial which is enough to make detection fail.

It should also be noted that our newly created song is much closer from the original than the marked version. Consequently, we cannot have any quality problems, and testing the quality of the final result is not required.

6.2 Going further

One can now argue that our attempt succeeded because the *same* pattern was used to mark both songs. First, we would like to point out that this is necessary because the detector needs to have this pattern in order to be able to work. If different patterns were used for every song, building a detector would be essentially impossible because it would have to test every possible patterns or recreate the pattern from the song. Recreating the pattern from the song, would require to solve either the problem of fuzzy hashing or of song classification, which are certainly both as hard as watermarking. Furthermore, even if song classification was realized, there would be a need to maintain a complicated database of all the existing songs together with their associated marks.

However, it is possible to use a small set of patterns, and it is also possible (and reasonable) that this pattern would not be the same in the version of the system deployed in real life.

We will now show that our attack still works. More precisely, we will show how to recover an unknown pattern from the marked song, *without* the original song.

Let us first assume, to simplify the exposition, that the mark starts at the first sample of the song. We use the same notation as in the rest of the article: s_i denote the $i-th$ chunk of 1470 samples of the unmarked song, and s'_i the corresponding chunks of the marked song, $w = (w_1, \ldots, w_{10})$ denotes the pattern (unknown here), and β denote the unknown function.

Let us also assume, again for simplicity, that the song s is exactly l chunks long. So, we have, for every i in $\{1, \ldots, l\}$, and every j in $\{1, \ldots, 10\}$,

$$s'_i[j] = s_i[j] + \beta(s, w, i, j)||s_i[j]||w[j] \tag{3}$$

Let us divide by $||s'_i[j]||$ and sum over i. We will use the following notations:

- $S'[j] = \sum_{i=1}^{l} \frac{s'_i[j]}{||s'_i[j]||}$
- $S[j] = \sum_{i=1}^{l} \frac{s_i[j]}{||s'_i[j]||}$

We have, for every j:

$$S'[j] = S[j] + w[j] \sum_{i=1}^{l} \beta(s, w, i, j) \frac{||s_i[j]||}{||s'_i[j]||} \tag{4}$$

The multiplicative term $\beta(s, w, i, j)\frac{||s_i[j]||}{||s'_i[j]||}$ is not very problematic. First, it turned out that it was extremely close to one for every j (actually, it would almost disappear if we knew β), second, is is not a real problem to recover the mark times a multiplicative constant. I would have been a problem if this sum happened to be very small, but that was not the case.

The more problematic term is $S[j]$. We would like it to be small. However, it is very difficult to estimate the typical value of $S[j]$. Naturally, if l is large enough, it should be very small. Having longer songs (the songs included in the challenge were only two minute long) would help. Also, if the same (now unkown) mark is used for several songs (which seems to be the case), we can actually perform the averaging on all the songs we can obtain, thus largely improving the chances for $S[j]$ to be negligeable.

The problem is that the structure of the music plays an important part in the value of $S[j]$. If, for example, a drum beat happens with a period which is synchronized with the period of the mark, $S[j]$ might be very large on some specific points.

We have not had time to perform an analysis of the value of $S[j]$ on a large number of songs, and we are not aware of a general statistical model for music. What we know, however, is that in the case of BM, our technique works surprisingly well. It turned out that the average of the unmarked version of this song was especially small. We could recover the mark from BM (and from *only* BM) with a very good precision. Figure 7 shows a part of the mark recovered from BM and the corresponding mark extracted from the difference $D = AM - A$. Once we have recovered the mark, the rest of the attack works as previously.

Note that, especially when $S[j]$ is not negligeable, it is possible to improve the precision on w by filtering $S'[j]$ to attenuate $S[j]$ in a very significant way. As a matter of fact, $S[j]$ and w are very different in nature. $S[j]$ is obtained by averaging the signal over periods of 147 samples and such a process is well known to be equivalent as low-pass filtering. On the contrary, the watermark w has the most important part of its information in the higher frequencies. Figure 8 illustrates very well this phenomenon on a different song than BM. Therefore, applying a high-pass filter with an adequate cutoff frequency (0.01 in the case of figure 8) allows the extraction of w with a higher precision.

As a final note, let us recall that we had assumed, for simplicity, that the mark was starting at the first sample of the song. This was not the case in BM. However, we simply needed to perform the averaging attack for every 1470 possible starting positions.

6.3 Generalization

It can now be argued that the attack was possible only because of the (approximate) knowledge of the function α, and that if a much more complicated function α had been used, the attack would have failed.

This is true. However, we would like to point out the following: as soon as we know α, even approximately, our attack works. Consequently, the security of a marking algorithm based on this type of scheme relies not only on the secrecy of the private key (the pattern), but also on the secrecy of the algorithm.

This is a problem for several reasons. First, conceptually, the cryptologic community considers extremely bad practice “security by obscurity”, that is, an algorithm whose security relies on the secrecy of its design. It has happened many times in the past that the details of an algorithm were divulgated by an unethical person.

Second, even if the secrecy can be maintained, there exists a very important practical issue: if a detector is distributed, either in software or hardware, it can be disassembled and analyzed, and it is considerably easier to protect a small piece of information, such as a key, than to protect the details of a full algorithm.

Consequently, we conjecture that if this system, or a system using the same core technology with a more sophisticated function α, is ever deployed, it will rapidly be defeated.

7 Conclusion

We have presented the analysis of one of the schemes proposed by the SDMI consortium, and have shown how to defeat it. We have also underlined that the security of many watermarking schemes rely not only on the secrecy of the mark but on the secrecy of the algorithm itself, and consequently that they are not suited for distribution.

References

[Auc98] David Aucsmith, editor. *Second Workshop on Information Hiding*, number 1525 in Lecture Notes in Computer Science. Springer-Verlag, 1998.

[CKLS96] I. Cox, J. Kilian, F. T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for images, audio and video. In *IEEE Int. Conference on Image Processing*, volume 3, pages 243–246, 1996.

[CMW$^+$] S. Craver, P. McGregor, M. Wu, B. Liu, A. Stubblefield, B. Swartlander, D.S. Wallach, D. Dean, and E.W. Felten. http://www.cs.princeton.edu/sip/sdmi/.

[HG96] Franz Hartung and Bernd Girod. Digital watermarking of raw and compressed video. In *SPIE 2952: Digital Compression Technologies and Systems for Video Communication*, pages 205–213, 1996.

[JDJ99] N.F. Johnson, Z. Duric, and S. Jajodia. Recovery of watermarks from distorted images. In Andreas Pfitzmann, editor, *Information Hiding Workshop '99*, Lecture Notes in Computer Science, pages 318–332. Springer-Verlag, 1999.

[otSg] Leonardo Chiariglione (Executive Director of the SDMI group). An open letter to the digital community. http://www.hacksdmi.org/letter.asp.

[PAK98] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Attacks on copyright marking systems. In David Aucsmith, editor, *Second Workshop on Information Hiding*, number 1525 in Lecture Notes in Computer Science, pages 218–238. Springer-Verlag, 1998.

[PAK99] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Information hiding. a survey. In *Proceedings of the IEEE, special issue on protection of multimedia content*, 1999. To appear.

[Pfi99] Andreas Pfitzmann, editor. *Third Workshop on Information Hiding*, number 1768 in Lecture Notes in Computer Science. Springer-Verlag, 1999.

[PP99] Shelby Perreira and Thierry Pun. Fast robust template matching for affine resistant image watermarks. In Andreas Pfitzmann, editor, *Information Hiding Workshop '99*, Lecture Notes in Computer Science, pages 199–210. Springer-Verlag, 1999.

A Figures

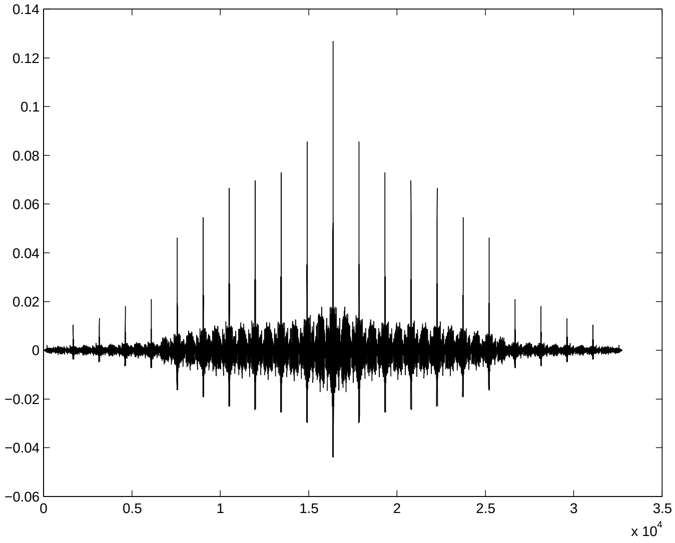

Fig. 2. The autocorrelation of *d*.

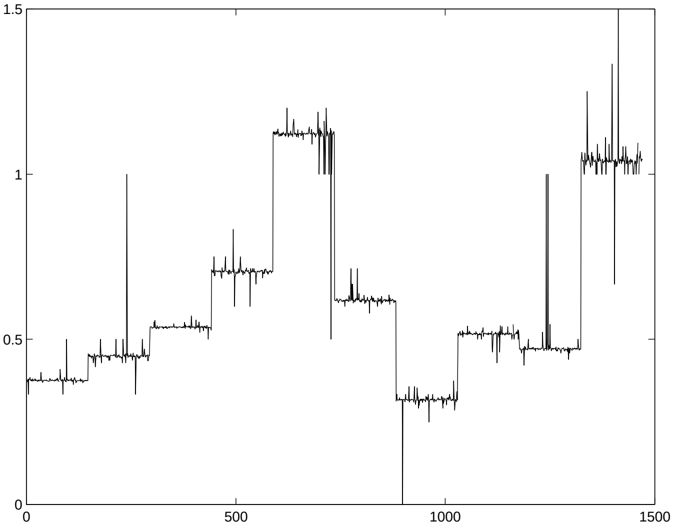

Fig. 3. The ratio of two successive periods of the mark.

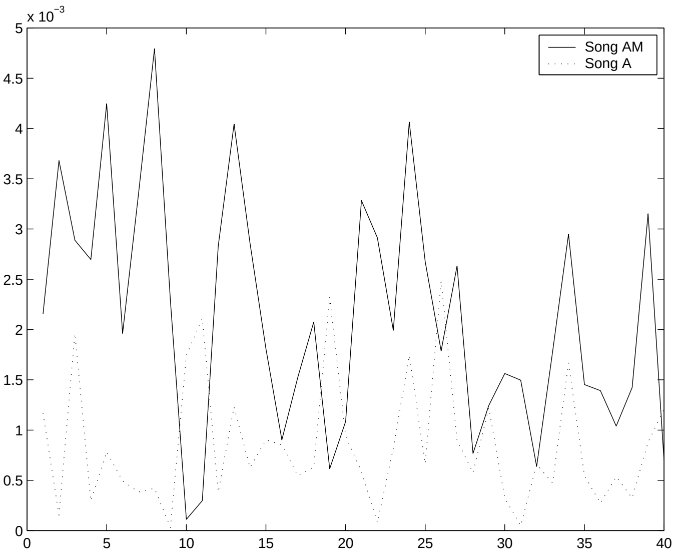

Fig. 4. Output of our detection algorithm with $p = 30$ on the original song A and its marked counterpart AM.

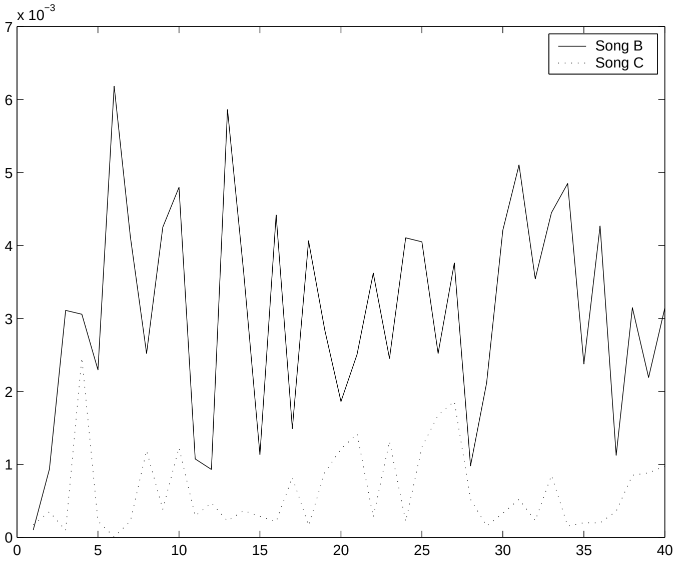

Fig. 5. Output of our detection algorithm with $p = 30$ on our newly produced song C and its marked counterpart B.

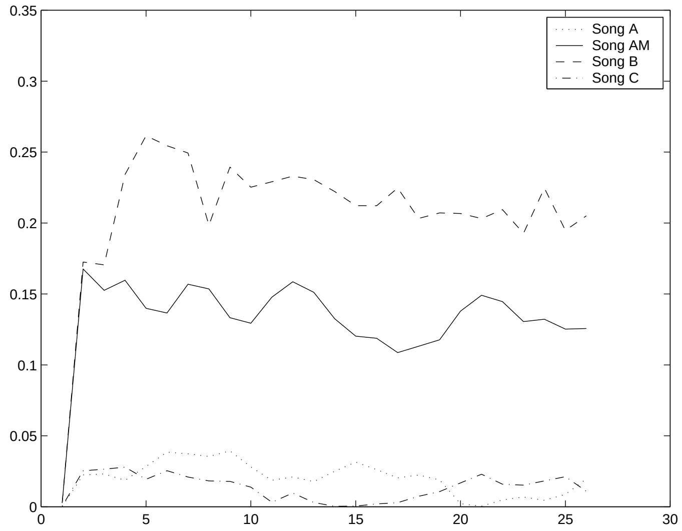

Fig. 6. Output of our detection algorithm with $p = 450$ on the four songs A, AM, B and C.

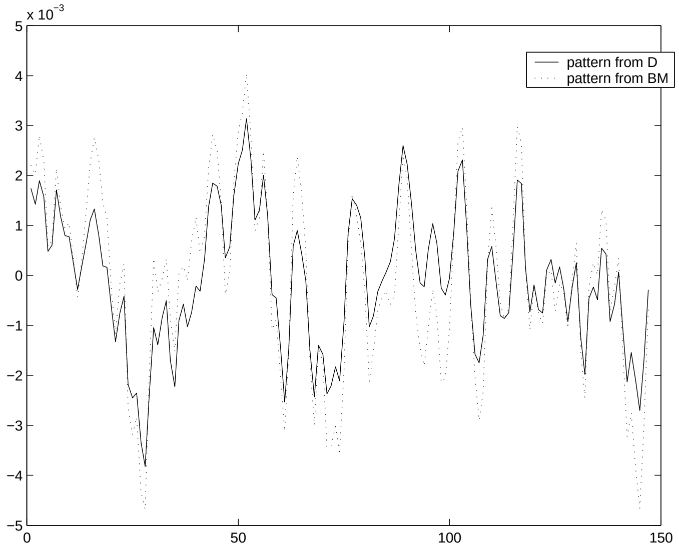

Fig. 7. Comparison between the mark recovered from the difference $D = AM - A$ and the mark recovered from BM.

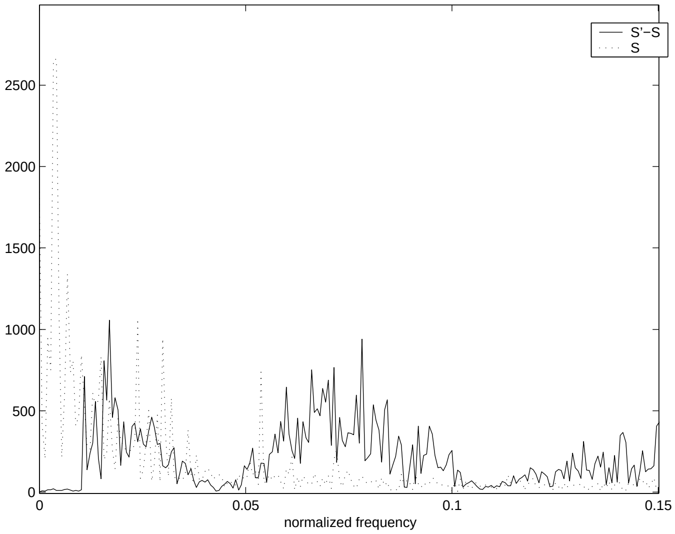

Fig. 8. Comparison between $S[j]$ and $S'[j] - S[j]$ in the frequency domain. Note that most of the energy of $S[j]$ is contained in the low frequency