FINGERING WATERMARKING IN SYMBOLIC DIGITAL SCORES

David Gross-Amblard
Le2i-CNRS Lab.
Université de Bourgogne
France
first.last@u-bourgogne.fr

Philippe Rigaux
Lamsade-CNRS Lab.
Université de Dauphine
Paris IX, France
first.last@dauphine.fr

Lylia Abrouk **Nadine Cullot**
Le2i-CNRS Lab.
Université de Bourgogne
France
first.last@u-bourgogne.fr

ABSTRACT

We propose a new watermarking method that hides the writer's identity into symbolic musical scores featuring fingering annotations. These annotations constitute a valuable part of the symbolic representation, yet they can be slightly modified without altering the quality of the musical information. The method applies a controlled distortion of the existing fingerings so that unauthorized copies can be identified. The proposed watermarking method is robust against attacks like random fingering alterations and score cropping, and its detection does not require the original fingering, but only the suspect one. The method is general and applies to various fingering contexts and instruments.

Keywords. Watermarking, fingering

1. INTRODUCTION

In this work we consider symbolic musical scores that contain *fingering annotations*. Such fingerings ease the score interpretation for the novice player, and can guide the professional player. Producing high quality fingerings is a complex and costly task for the score writer. Up to now, it mainly remains an hand-made task, although several automatic fingering methods have been proposed recently [1–3].

The score writer's investment is threaten by the development of musical scores in digital form. Any buyer of such scores can obtain a perfect copy of the files and resell illegal copies. Watermarking is a known tool to protect the intellectual property of digital content, and it can be envisioned for musical scores as well. This would enable the distribution and sharing of score files marked by the copyright of their owner(s), just like score sheets are nowadays, but with the numerous advantages associated with the digital format.

Several methods have been proposed to hide the owner's identity into score images, by changing pixels [4], staff thickness [5] or symbols shape [6, 7]. These approaches are well fitted for protecting score images, but are not relevant for data exchange in a symbolic format like MusicXML [8]. Given the high cost of producing a symbolic digital score, writers may demand a robust mechanism to embed their copyright mark in the music symbolic representation. This copyright mark must be preserved throughout the operations that can be applied to the digital representation (e.g., transposition). It should not depend on side aspects such as graphical output details (e.g., the thickness of staff lines) which can easily be replaced or even eliminated without harm, as they are not part of the symbolic representation. Finally, the watermark should not alter the music content. In order to satisfy these requirements, our approach consists in watermarking the existing scores annotations. In the present paper we apply this idea to fingering annotations. Up to our knowledge, this is the first work on watermarking the music semantics itself.

The key idea of the method, given a musical score and a hand-made high quality fingering, is to choose several short secret fragments of the score. Given a score fragment, we replace the existing fingering with another fingering, chosen secretly among several computer-made fingerings of comparable quality. All secret choices are made using a cryptographic pseudo-random number generator, seeded by a summary of the musical structure and with a secret key known only by the legitimate owner. The resulting fingering will be published with the musical score. Finally, given a suspect score, the correspondence of the suspect fingering with our secret choices on our secret fragments acts as the proof of ownership. Our method applies to any fingering scenario, as soon as a quality metric of fingerings is available along with an automatic fingering method for small fragments (such as in piano or guitar music for example).

It should be clear that we protect the combination of the score and its fingering, and not the score itself. We also suppose that the attacker cannot afford to alter the score significantly, as this would result in an unsellable score (nevertheless we moderate this assertion in Section 3).

Outline. The paper is organized as follows. In Section 2 we introduce our general model for fingering and watermarking. Section 3 presents our watermarking and detection algorithms. Section 4 discusses several issues on the robustness of the watermark against natural score manipulation or malevolent attacks. Experiments assessing our method are presented in Section 5. Section 6 briefly covers the related work and Section 7 concludes.

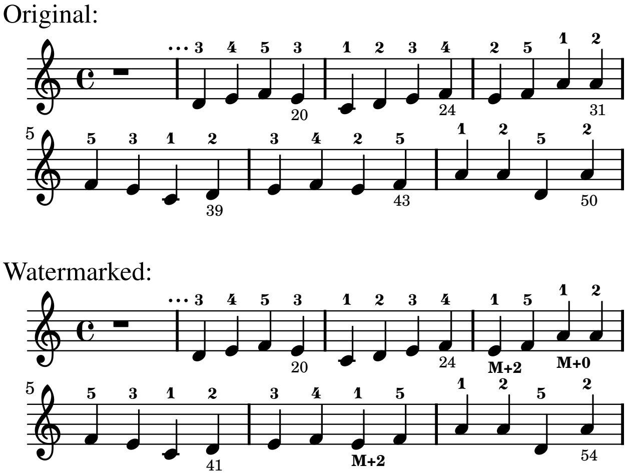
2. FINGERING AND WATERMARKING

2.1 Fingering

The method proposed in this paper applies to any fingering context, but for the sake of simplicity we will focus on right-hand piano fingering for melodic inputs. Given a score in symbolic notation, we abstract it as a sequence $s = (n_1, \ldots, n_N)$ of N consecutive notes. A fingering $f(n_i)$ for a note n_i is an integer in $\{1, 2, \ldots, 5\}$, where number 1 to 5 represents a right-hand finger, respecting the usual conventions. For example, $f(A) = 2$ means that note A will be played by the forefinger.

The watermarking method uses an estimate of the quality of a fingering, that is related to the player inner feelings. We suppose the existence of a cost function $cost(f, s)$ that provides the cost of fingering f for the score s: the higher the cost output by this function, the lower the quality of the provided fingering (such functions exist for several instruments like piano [1]). We will explicit such a function in the experiments of Section 5, but our method applies to any cost function. We also often use the cost of a fragment w of the score s, that we denote $cost(f, w, s)$.

The first staff of Figure 1 presents an original score fragment with fingering annotations built by the score writer. Fingering annotations appear above the score. Annotations below the score are presented here only for the purpose of explanation, but are not published by the score writer. They show the cumulative cost of playing the score with the corresponding fingering (for example, playing the whole score costs 50 according to the chosen cost function).



Original:

Watermarked:

Figure 1. Different fingerings of the same score, with cumulative costs

2.2 Watermarking protocols

A watermarking protocol is a pair of algorithms $(\mathcal{W}, \mathcal{D})$, where $\mathcal{W}$ and $\mathcal{D}$ are respectively the marker and detector algorithms (see Figure 2). Given an original score s and a high quality fingering f, the score writer will watermark it by obtaining a specific fingering $f_M = \mathcal{W}(s, f, \mathcal{K})$, depending on a secret numerical key $\mathcal{K}$. The watermarked score (s, f_M) is sold to users. If a suspect copy (s^*, f^*) is discovered, the detector $\mathcal{D}$ applied on (s^*, f^*) using the secret key $\mathcal{K}$ should output $guilty$ if f^* was obtained from f_M, and $not\ guilty$ if f^* is a fingering obtained independently from f_M. A watermarking protocol is said to be *blind* if the original fingering is not needed at detection time, which may be useful as writer's fingerings may not be accessible easily or archived properly. The suspect fingering may have been also attacked/distorted before reselling, in order to erase the watermark. A watermarking protocol is said to be *robust* if it can still detect reasonably altered fingerings. Finally, respecting usual conventions, marker and detector algorithms are public, and their security relies only on the secret key.

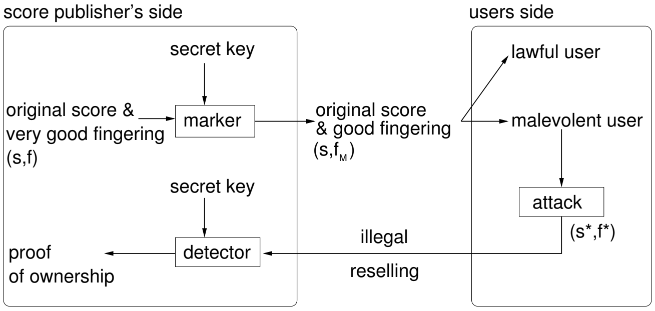


Figure 2. Protecting score and fingering by watermarking

3. FINGERING WATERMARKING

3.1 Watermarking algorithm

Algorithm 1 gives the pseudo-code of the marker. This algorithm scans a given score s by considering only a window of k consecutive notes (line 4 and 5). For each window, we first decide if it constitutes a good candidate for watermarking (line 6 and 7). This choice is secret and is based on the window content, the secret key $\mathcal{K}$ and a watermarking period γ known only by the score writer (this will be explained in the next section).

If a given window w is considered for watermarking, we focus on its first note n_i. We try to replace the original fingering $f(n_i)$ for this note by another one, $f'(n_i)$, also chosen secretly between the 5 possible fingerings for our piano example (line 9).

We compare the cost of this new fingering $cost(f', w, s)$ on window w with the cost of the original fingering $cost(f, w, s)$ on w (line 10). If the new cost exceeds the previous one by a limit ε, we cancel this modification (line 12). If the new fingering has a reasonable cost, we keep it for publication. Parameter ε, chosen by the score writer, controls the allowed amount of alteration that results from the watermarking process, and guarantees to produce fingerings with a good quality.

The second staff on Figure 1 demonstrates the process. For example, the 9th note (E) is considered for watermark-

```
Algorithm 1: Watermarking
   Input: a score $s$ of $N$ notes $n_1, \dots, n_N$, a high
          quality fingering $f$ for $s$, a secret key $\mathcal{K}$, a
          window size $k$, a quality threshold $\varepsilon$, a
          period $\gamma$.
   Output: a watermarked fingering $f'$.
1  begin
2    // copy $f$ to $f'$
3    $f' := f$
4    for $i = 1$ to $N - k + 1$ do
5      $w = n_i.n_{i+1} \dots n_{i+k-1}$ // reference window
6      seed PRNG $G$ with $signature(w).\mathcal{K}$
7      if ($G.nextInt() \mod \gamma = 0$) then
8        // try to watermark the first note
9        $f'(n_i) := G.nextInt() \mod 5$
10       if ($|cost(f', w, s) - cost(f, w, s)| > \varepsilon$) then
11         // revert changes
12         $f'(n_i) := f(n_i)$
13       end
14     end
15   end
16   return $f'$
17 end
```

ing. Its original fingering (finger 2) has been replaced by a new fingering (finger 1). This yields an extra cost of 2, which is considered reasonable for this example. The overall watermarking process yields a total extra cost of 4 on the score fingering.

3.2 Randomness

We now explain how random choices are made. Given a window w, we compute its musical signature based on its core music content (*signature()* function, line 6). The signature is independent from annotations and ornaments that are pointless for our algorithm. It is robust against naïve transposition attacks as it transposes the score into a common key (but of course, fingering costs are computed according to the original score). It is also invariant against score rewriting replacing a note or group of notes by an equivalent encoding (for example, replacing a half note by two tied quarters). In this paper, the signature is the concatenation of transposed note pitches, where consecutive equal pitches are suppressed. For example, the signature of `ABAABC` is `ABABC` (seen as a number), and time is not taken into account.

We concatenate this signature with the secret key $\mathcal{K}$ (a number), known only by the score writer. Then (line 6), we seed a cryptographic pseudo-random number generator (PRNG) with this number (as in [9]). This generator is used for all subsequent choices and has interesting properties. First, if it is seeded with the same value, the produced numbers are deterministic. Hence, if we know the secret key, we will be able to reproduce the pseudo-random choices made at watermarking time. Second, if the secret key is unknown, the generator outputs look completely random and can not be reproduced. Hence an attacker, unaware of the secret key, is fighting against randomness.

3.3 Detection algorithm

```
Algorithm 2: Detection
   Input: a suspect score $s$ of $N$ notes $n_1, \dots, n_N$ with
          its fingering $f^*$, a secret key $\mathcal{K}$, a window size
          $k$, a quality threshold $\varepsilon$, a period $\gamma$, a security
          parameter $\delta$.
   Output: $guilty$ or $not\ guilty$.
1  begin
2    // copy $f^*$ to $f'$
3    $f' := f^*$
4    $total := 0$, $match := 0$
5    for $i := 1$ to $N - k + 1$ do
6      $w = n_i.n_{i+1} \dots n_{i+k-1}$ // reference window
7      seed PRNG $G$ with $signature(w).\mathcal{K}$
8      if ($G.nextInt() \mod \gamma = 0$) then
9        // check this window
10       // compute awaited value
11       $f'(n_i) := G.nextInt() \mod 5$
12       if ($|cost(f', w, s) - cost(f^*, w, s)| \le \varepsilon$) then
13         // probably watermarked position
14         $total$++
15         if ($f'(n_i) = f^*(n_i)$) then
16           $match$++
17         end
18       end
19       else
20         $f'(n_i) := f^*(n_i)$ // revert changes
21       end
22     end
23   end
24   if ($match/total > \frac{1}{5} + threshold(N, \delta)$) then
25     return $guilty$
26   else
27     return $not\ guilty$
28   end
29 end
```

The detection algorithm (see Algorithm 2 for the pseudocode) proceeds like the marker algorithm. Using the same window size, watermarking period and secret key used at watermarking time, we seed the generator with each window signature and the secret key (line 7). Hence, the same random choices made at watermarking time are reproduced. Thus we can locate exactly those windows selected at watermarking time (line 8). Then, *since the detector does not have the watermarked fingering for comparison* (blind detector), we have to assess that this position has really been used for watermarking. For that, we replace the fingering of the first note by the awaited one, using the random generator (line 11). We then compute the cost of this fingering. If it exceeds the error limit ε, we discard this window and restore the initial fingering (line 20). If error limit is respected, this position is probably a watermark (line 14).

We then compare the awaited fingering with the found one (line 15). For the whole score, we maintain the ratio of the number of matching fingerings with the number of windows considered for detection. If this ratio exceeds a given threshold (line 24), we consider the score as suspect (the threshold value is discussed below).

4. DISCUSSION

In this section we discuss several classical issues related to watermarking algorithms.

Impact on quality. Since the PRNG outputs random numbers with uniform distribution, the probability for a window w to be considered for watermarking is $1/\gamma$. The impact of watermarking this window can not be higher than ε. Hence, for a N notes score, the mean overall alteration is at most $\varepsilon\lfloor N-k\rfloor/\gamma$.

Window size. As the window size k increases, the amount of randomness injected into the random generator extends. If we consider reasonable scores whose notes span 2 octaves, there is up to 14^k potential fingerings for k consecutive notes. We chose $k=5$ in our experiments, leading to half-a-million distinct window signatures.

False positives probability and threshold function. A false-positive detection occurs when the detector considers a random score as guilty. Clearly, this probability must be negligible. Let δ be this acceptable probability, say $\delta = 10^{-10}$. Let us consider a random score. The probability of a given window to be selected by the detector is $1/\gamma$. For piano fingering, the probability of a fingering to correspond – by chance – to the watermarked one is 1/5 (as there is 5 different possible fingerings). Hence the average number of total matches on a random score is $\lfloor N-k\rfloor/5\gamma$. By the Hoeffding bound [10], the probability that the detector ratio $\frac{match}{total}$ on a random score deviates from the previous average is such that

$$P[|\frac{match}{total}-\frac{1}{5}| > threshold(N,\delta)] < e^{-2\frac{N}{\gamma}threshold(N,\delta)^2}.$$

Hence, choosing $threshold(N,\delta) = \sqrt{\frac{\gamma}{N}\ln\frac{1}{\delta}}$ guarantees a false positive rate smaller than δ. For example, on a score of 10,000 notes with a watermarking period $\gamma = 10$ and $\delta = 10^{-10}$, the recommended threshold is 0.22.

Available bandwidth. Robustness and significance are proportional to the amount of watermark bits that can be hidden. In popular guitar pieces (e.g., guitar scores and tablatures for beginners), a significant number of watermark positions are available. But music for expert players may contain only a few fingering annotations. If this number is not sufficient to reach the security limit, or if the musical corpus is made of small pieces only, a natural extension is to consider the watermarking of an entire piece collection (collected in a CD for example). The watermark is spread on the collection, and since the detection method uses only a finite-size sliding window, the order of pieces within the collection is pointless at detection time. The method is also robust enough to recover the watermark on a subset/superset of scores.

Attacks. An attacker suspecting the occurrence of a watermark may try to evade detection by several means. First, the attacker can add easy-to-correct errors in the fingering. To be successful, the attacker will have to add such errors all along the piece, in order to erase sufficient watermark positions. Hence the overall fingering is full of errors. Second, the attacker can leave the fingering unchanged, but add errors on the score itself, in order to break synchronization with the fingering. If errors are simply equivalent notes rewritings, the signature method will probably recover the correct ones. If the error is large, it will break one watermark position. Again, errors must span the whole score to be efficient, which is unreasonable (due to lack of space, we omit the mathematical proof of these statements. They are similar to the false-positive analysis).

Another approach for the attacker is to refinger the score. A complete rewriting represents a significant amount of work, so why would this attacker bother buying a fingered score in the first place ? On the contrary, a small refingering acts as a random attack, as the attacker has no idea where to perform this fingering.

Finally, the malevolent user can attack the score structure. Brute-force transposition is not sufficient, as we normalize the score in a specific key for detection. A first technique is to resell only subscores (excerpts). This can occur even for a normal buyer using the score. However, as long as a significant fraction of the piece is present, the watermark can be detected (this fraction is typically 30% in the database watermarking literature [9]). If less than 1/3 of the piece is stolen, the loss of property is harmless. If an attacker mixes a watermarked collection with a huge number of unwatermarked pieces, the argument is similar.

A last technique is to fold or unfold the score according to repetition symbols. This attack can be counterfeited by discarding repeated parts in the $signature()$ function, both for watermarking and detection.

5. EXPERIMENTS

5.1 Data, cost function, parameters

Our experiments are based on 50 Chopin piano pieces from the KernScores repository [11], for a total of around 10,000 notes. Original fingerings were found with a Dijkstra algorithm using a fingering cost function close to [1] and [2] (our method supposes hand-made high quality fingerings, but this approach is sufficient to measure the watermarking impact on quality). These models encompass the cost of playing a note with a given hand position (vertical cost $cost_v(f,n)$), and the cost of the transition between one hand position to the next one (horizontal cost $cost_h(f_i,n_i \rightarrow f_{i+1},n_{i+1})$). These costs are constant values that agree with the human hand physical possibilities (the precise definition of these costs in not relevant for the present paper, we refer the reader to [2] for in-depth explanations.) The $cost(f, n)$ of a fingering f is the sum of its horizontal and vertical costs, i.e.,

$$cost(f, n) = \sum_{i=1}^{N} cost_v(f_i, n_i) + cost_h(f_i, n_i \rightarrow f_{i+1}, n_{i+1}).$$

We used window size $k = 5$, error tolerance $\varepsilon = 10$ and detection threshold 0.8 (vertical and horizontal costs for one note or transition span between 0 and +14).

5.2 Experiments

Figure 3 shows the impact of the watermarking method for various values of watermarking period γ. Clearly, a period smaller than 5 yields a huge distortion, and greater values tend toward a constant error with respect to the original fingering. Figure 4 and 5 study the impact of a random attack that tries to erase the watermark as follows: a note fingering is chosen with probability $1/\gamma_a$, and changed into a random fingering up to a cost impact of 10. Figure 4 shows the attack impact on the watermarked fingering quality for various values of γ_a. It appears that the attack impact is larger than the watermark impact on the fingering cost: choosing $\gamma_a < 5$ leads to fingerings with poor (unsellable) quality. Figure 5 shows the attack impact on the detector ratio. Choosing a detection threshold of 0.8 guarantees that all suspect fingerings are correctly detected, expect for those with attack γ_a smaller than 6. Hence, Figure 4 and 5 argue that any attack tricking the detector also destroys the fingering quality. Finally, Figure 6 shows that using a random secret key does not yield false positive detection (the correct key is presented at index 50).

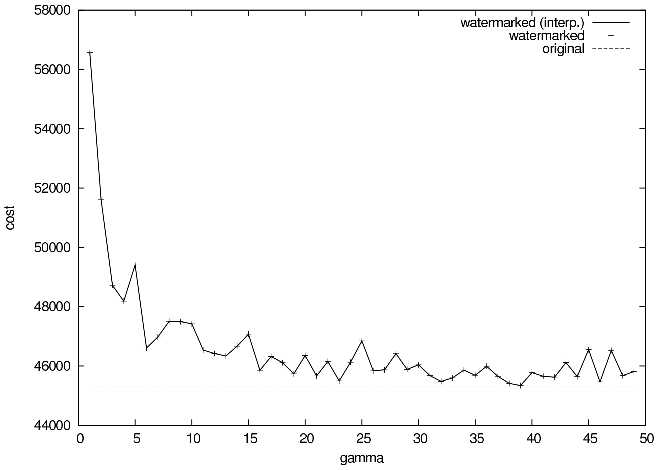


Figure 3. Impact of watermarking on fingering cost

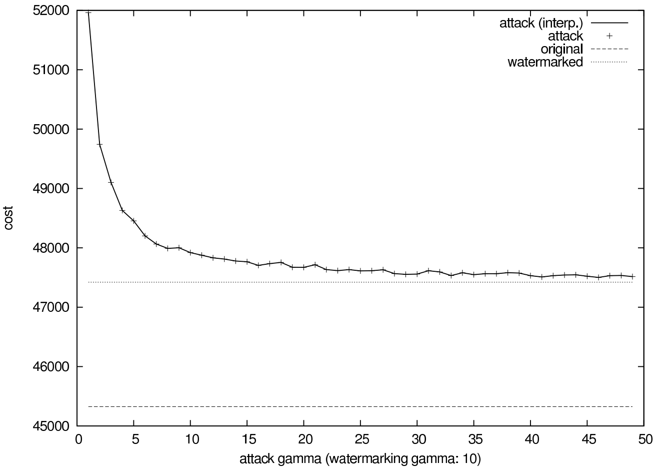


Figure 4. Impact of attack on fingering cost

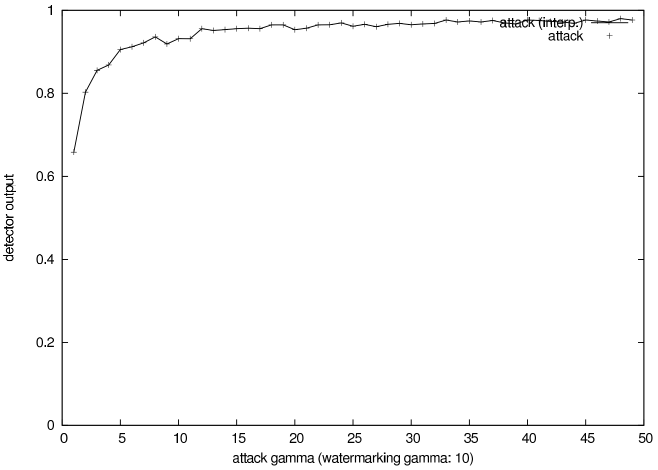


Figure 5. Impact of attack on detector's output

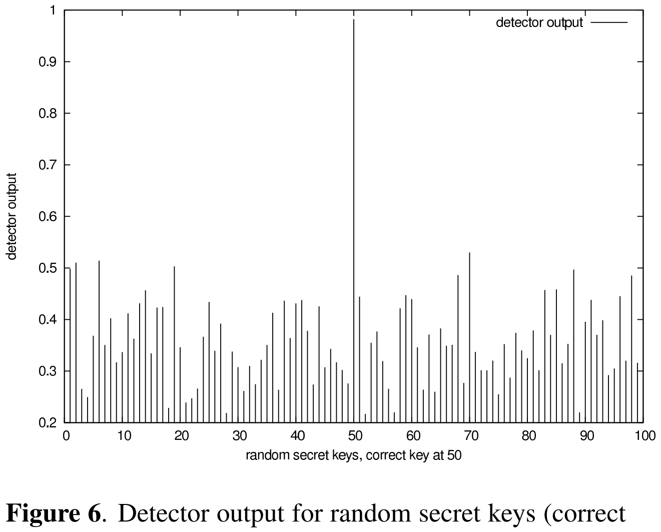


Figure 6. Detector output for random secret keys (correct one at 50)

6. RELATED WORK

Hiding information (for various purposes) in musical scores is an old story. A study of music score watermarking was performed during the WEDELMUSIC project. A good survey [12] recalls these approaches. In the visual domain, classical but adapted image watermarking techniques can be applied on the image of a musical score. The watermark can be hidden by altering grayscales, or the binary representation of images, or the pixels themselves. In the musical notation (but still into the score image), one can alter the staff thickness, the vertical or horizontal distance between notes or groups of notes, notes orientation, thickness [5] or shape [6, 7]. Little is known on information hiding into the music semantics, where our work stands.

Our method shares some similarities with database watermarking methods: watermarking of relational databases of numerical values [9], numerical data streams [13] and XML streams [14]. All these methods use the same PRNG technique, and [13, 14] also use a finite window to scan a numerical or textual stream. The main difference is that our method has to control a non-local cost on data and may require rollbacks.

7. CONCLUSION

On-line distribution of musical scores is a promising area. Among other advantages, it could offer instant access to music collections, a wide diffusion of rare musical pieces, and computer-based services to browse, recommend, search and analyze music. However, producing music scores is a costly process and the protection of score writers against illegal copies is a prerequisite for on-line collection to emerge. In the present paper, we propose a watermarking algorithm based on the idea that the owner signature should be based on the musical content (which can hardly be modified) and hidden in a valuable annotation of this content – namely, fingerings. We propose a simple algorithm and show that it results in an effective protection. Although currently limited to fingerings, we believe that our approach can be extended to music annotations in general, for instance lyrics in vocal music. We are currently investigating this larger context.

8. ACKNOWLEDGEMENT

This work is partially supported by the French ANR Content and Interaction *Neuma* Project [15].

9. REFERENCES

[1] Melanie Hart, Robert Bosch, and Elbert Tsai. Finding optimal piano fingerings. *The UMAP Journal*, 2(21):167–177, 2000.

[2] Alia Al Kasimi, Eric Nichols, and Christopher Raphael. A simple algorithm for automatic generation of polyphonic piano fingerings. In *International Society for Music Information Retrieval Conference (ISMIR)*, pages 355–356, 2007.

[3] Yuichiro Yonebayashi, Hirokazu Kameoka, and Shigeki Sagayama. Automatic decision of piano fingering based on a hidden Markov model. In Manuela M. Veloso, editor, *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2915–2921, 2007.

[4] Wolfgang Funk and Martin Schmucker. High capacity information hiding in music scores. In Paolo Nesi, editor, *First International Conference on WEB Delivering of Music, proceedings of Wedelmusic 2001*, number 5020 in IEEE Computer Society Technical Committee on Computer Generated Music, pages 12–19, Los Alamitos, California, USA, 2001. IEEE Computer Society.

[5] Martin Schmucker. Capacity improvement of a blind symbolic music score watermarking technique. In Wah Ping Wong, editor, *Security and Watermarking of Multimedia Contents IV.*, number 4675 in SPIE Proceedings, pages 206–213, Washington, 2002.

[6] Martin Schmucker and Hongning Yan. Music score watermarking by clef modifications. In Edward J. Delp, editor, *Security and Watermarking of Multimedia Contents V.*, number 5020 in SPIE Proceedings, pages 403–412, Bellingham, 2003.

[7] Martin Schmucker, Christoph Busch, and Anoop Pant. Digital watermarking for the protection of music scores. In Wah Ping Wong, editor, *Security and Watermarking of Multimedia Contents III*, number 4314 in SPIE Proceedings, pages 85–95, Washington, 2001.

[8] Recordare. MusicXML document type definition. `http://www.recordare.com/xml.html`.

[9] Rakesh Agrawal, Peter J. Haas, and Jerry Kiernan. Watermarking relational data: framework, algorithms and analysis. *VLDB J.*, 12(2):157–169, 2003.

[10] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

[11] Craig Stuart Sapp. Online database of scores in the Humdrum file format. In *International Society for Music Information Retrieval Conference (ISMIR)*, pages 664–665, 2005.

[12] M. Monsignori, Paolo Nesi, and Marius B. Spinu. Watermarking music sheets. In *PCM '01: Proceedings of the Second IEEE Pacific Rim Conference on Multimedia*, pages 646–653, London, UK, 2001. Springer-Verlag.

[13] Radu Sion, Mikhail J. Atallah, and Sunil Prabhakar. Rights protection for discrete numeric streams. *IEEE Trans. Knowl. Data Eng. (TKDE)*, 18(5):699–714, 2006.

[14] Julien Lafaye and David Gross-Amblard. XML streams watermarking. In *IFIP WG 11.3 Working Conference on Data and Applications Security (DBSEC)*, 2006.

[15] Neuma project: network-enabled and user-friendly music analysis tools, 2008. `http://neuma.irpmf-cnrs.fr`.