Summary of "Fingerprinting IPs Using Constraint-Addition: Approach and Graph Case Study"

Ginger Myles

March 4, 2002

1 Introduction

Intellectual property (IP) protection is an important issue for people who design products such as software. Because of this importance many techniques have been explored for the use in IP protection. Once such technique is that of watermarking. It turns out that watermarking has some major limitations for use with IP. These limitations will be explored later. The authors of this paper have proposed a technique which they feel better addresses the neeeds of IP protection. The proposed technique is to use fingerprinting, but with a slightly different flavor so as to reduce runtime overhead and control the number of distinct fingerprinted copies produced.

Two of the major difficulties in IP protection is that the functionality must be preserved. In protecting things like audio or video the introduction of small errors is acceptable as long as it is not preceivable by the human auditory or visual system. With IP the introduction of errors could change the functionality. The second difficulty pertains to the illegal redistribution of the IP. If the same mark is placed in all copies of the IP it is not possible to identify the guilty party. So the goal is to find a way to place a distinct mark in every copy of the IP that can be detected but that does not change the functionality.

2 Watermarking Techniques

As was previously mentioned the protection of multimedia data is an easier task than that of IP protection. This is due to the fact that we can make use of the limitations of the human visual and auditory system. We are not as fortunate with IP because the functionality must be maintained. One watermarking method that has been tried on IP is constraint-based watermarking. Constraint-based watermarking takes the signature that needs to be embedded into the IP and makes it into an additional constraint that must be met during the design and implementation of the IP. The effectiveness of this technique is explored in another paper by Qu [6], which also makes use of the Graph Coloring problem.

3 Previous Fingerprinting Techniques

Fingerprinting is basically a type of watermarking where each watermark is unique, just like human fingerprints. In general the way fingerprinting works is to introduce errors into each copy of the

IP that will make that copy unique. These errors are called marks and it is the combination of these marks that comprise the fingerprint. Fingerprinting techniques are characterized as either symmetric or asymmetric. With symmetric fingerprinting both the buyer and the merchant know the data with the fingerprint. This type of fingerprinting does not make it possible to prove illegal redistribution by the buyer. In asymmetric fingerprinting only the buyer knows the data with the fingerprint. Neither of these techniques are suitable for the protection of IP.

4 Fingerprinting for IP Protection

In fingerprinting a set of marks is selected and each mark is a single bit of information. When embedding the fingerprint for each mark either a 0 or 1 is selected thus forming a binary word. There are two general assumptions made about the item being watemarked:

- Error-tolerance assumption: the object should remain useful after introducing small errors, and the user cannot detect the marks from the data redundancy.
- Marking assumtion: two or more users may detect a few marks that differ in their copies, but they cannot change the undetected marks without rendering the object useless.

In order for a fingerprinting technique to be effective it must meet the following criteria:

1. The protocol, with the help of the existing problem solver, should be able to create $K >> 1$ solutions at the expense close to that for finding one single solution.
2. The protocol, combined with the problem solver, should be capable of generating solutions that are far away from each other, otherwise collusion-secure solution distribution schemes are not possible.
3. The fingerprinting protocol should be non-intrusive to the problem solver so that it can be transparently integrated with existing design flows via pre-processing and/or post-processing.
4. The fingerprints must provide high credibility to identify each and every user.
5. The overhead or degradation of the quality of solution after embedding fingerprints should be minimized.
6. The user cannot remove or alter the fingerprints without complete knowledge of the IP or rendering the IP useless.
7. The IP provider can, without much difficulty, trace the guilty user(s) from the fingerprints in illegal copies.

In this paper IP protection is tackled using constraint-addition fingerprinting. The idea behind this approach is to make use of solutions generated from the Graph Coloring problem in order to fingerprint IP. The main difference between the technique used in this paper and in others is that the solutions are generated with as little overhead as possible. The purposed fingerprinting procedure consists of two phases: solution generation and solution distribution. One of the difficulties with this type of technique is making sure enough solutions are generated so as to assign a unique fingerprint to each copy. A way to deal with this problem is to use an NP-complete problem, like graph coloring or satisfiability. Because these problems are hard to begin with, generating enough solutions is also hard so extra constraints are applied to the initial problem which allows many solutions to be obtained from one solution, hence the name constraint-addition. In addition to the generation of solutions being difficult so is maintaining a one-to-one mapping from the solution to the copy. Since each solution is generated from a specific base, each solution can be uniquely expressed from this base. Thus each user's signature is mapped to a set of coefficients that correspond to the solution used.

5 Using Graph Coloring

The optimization graph coloring problem is: given a graph $G(V, E)$ color the graph with as few colors as possible such that no two adjacent vertices are colored with the same color. This NP-complete problem is used to generate the solution space for the fingerprinting. In the article four techniques are presented for generating multiple solutions from the one base solution.

The first technique is vertex duplication. This approach is based upon the idea that if we have one coloring scheme for the graph and we know that a particular vertex can also be colored with another color, then we have just found a second solution. The idea is to select a vertex and dublicate it. This new vertex is connected to the same vertices as the orignal and it is connected to the original to ensure that it is colored with a different color. Now there are two solutions for the same graph. This idea is applied to each vertex in the graph. Thus, if k vertices can be colored by two different colors there are now 2^k solutions. In this technique the k vertices which can be assigned an alternate color along with the colors serve as the base for the solution space.

The second technique is clique manipulation. A clique is a set of vertices where each vertex is connected to everyother vertex in the set. In clique manipulation, a clique of size k is chosen and for each vertex in the clique its neighbors are connected to all other vertices in the clique. This approach allows for the generation of $k!$ different solutions simply by assigning each vertex in the clique of the k colors. For this technique the base of the solution space is the combination of the different cliques selected.

The third technique is bridge construction. In this approach a pair of unconnected vertices is selected. Then one vertex is connected to each neighbor of the other vertex and the vertices themselves are connected. After the graph has been colored the two vertices will have different colors, so four solutions can be generated for the original graph. This method can also be extended to k unconnected vertices so k^k different solutions can be generated. This approach is based upon the watermarking technique where a signature is embedded into a graph by adding edges between pairs of unconnected vertices.

The last technique is solution post-processing. In this technique there is a graph $G(V, E)$ which is k-colorable. The set V_i is a subset of V in which all of the vertices have been colored by the i^{th} color. l-colors are selected to form the set $V' = \cup_{j=1}^{l} V_{i_j}$ of vertices. It is known that the subgraph $G[V']$ is l-colorable. If $G[V']$ is small enough then the solutions can be generated by brute force. A second graph $G[V'']$ can be generated in the same way, but $V' \cap V'' = \emptyset$. Thus, if n_1 and n_2 solutions have been found for the two subgraphs then $n_1 \times n_2$ solutions have been found for the original graph.

6 Solution Distribution Scheme

In deciding on a distribution scheme it is necessary to try to increase the difficulty in determining the entire solution space from compairing copies. The idea used in this paper is to select a subset of the solutions generated and use only solutions from that subset. This makes it difficult for users to generate new solutions which could be used for illegal redistribution of the IP. Creating just any random subset is not enough instead the subset should be selected carefully and should satisfy the following requirements:

- Any combination of solutions cannot create a new solution in the subset
- From any solution created by a combination of solutions, at least one of the original solutions can be traced.

Following this distribution scheme will make it easier to determine who is guilty in the case of illegal redistribution of IP.

7 Results

The fingerprinting technique purposed in the paper was tested on both random graphs and real-life benchmark graphs. It was found in the experiments that fingerprinting the random graphs introduced overhead, but the fingerprinting of the reallife graphs did not suffer from run-time overhead.

8 Conclusion

The authors outlined 5 main benefits that they claim their approach provides over previous IP protection techniques.

1. Run-time overhead for generating many solutions is almost zero.
2. In three of the four techniques that were implemented, the number of solutions can be controlled and the solutions are guarenteed distinct.
3. The IP provider's signature can be embedded in the fingerprinting process without additional watermarking techniques.
4. Both symmetric and asymmetric fingerprints can be created.
5. With proper distribution schemes, the techniques are collusion-secure.

From the evidence provided it appears as if the authors have supported these claims and that the fingerprinting will work for the protection of IP.

References

[1] Andrew E. Caldwell, Hyun-Jin Choi, Andrew B. Kahng, Stefanus Mantik, Miodrag Potkanjak, Gang Qu, and Jennifer L. Wong. Effective iterative techniques for fingerprinting design IP. In *Design Automation Conference*, pages 843–848, 1999.

[2] Andrew B. Kahng, John Lach, William H. Mangione-Smith, Stefanus Mantik, Igor L. Markov, Miodrag Potkonjak, Paul Tucker, Huijuan Wang, and Gregory Wolfe. Watermarking techniques for intellectual property protection. In *Design Automation Conference*, pages 776–781, 1998.

[3] F. Koushanfar, G. Qu, and M. Potkonjak. Intellectual property metering. In *4th Information Hiding Workshop*, pages 87–102, 2001.

[4] G. Qu, J.L. Wong, and M. Potkonjak. Fair watermarking techniques. In *EEE/ACM Asia and South Pacific Design Automation Conference*, pages 55–60, 2000.

[5] Gang Qu and Miodrag Potkonjak. Analysis of watermarking techniques for graph coloring problem. pages 190–193.

[6] Gang Qu and Miodrag Potkonjak. Hiding signatures in graph coloring solutions. In *Information Hiding*, pages 348–367, 1999.

[7] Gang Qu, Jennifer L. Wong, and Miodrag Potkonjak. Optimization-intensive watermarking techniques for decision problems. In *Design Automation Conference*, pages 33–36, 1999.