Secret key watermarking with changing keys

SECRET KEY WATERMARKING WITH CHANGING KEYS

Geert Depovere and Ton Kalker

Philips Research
Prof. Holstlaan 4 (building WY82)
5656 AA Eindhoven, The Netherlands
Tel: +31 40 2743042, Fax: +31 40 2744675
Email: Geert.Depovere@philips.com

ABSTRACT

In this paper we consider a digital watermarking application where multiple parties can embed additional information using their watermark embedder. These parties are not supposed to influence each other and each watermark detector needs to be able to decode the information embedded by any of the embedder systems.
One approach would be to use a single secret key and to assign part of the payload to identify the particular embedder. However, it is generally accepted that for security reasons, each embedder should better have its own secret key. A major drawback of this last approach is related to the detector implementation complexity, which increases linearly with the number of embedders.
In this paper it is shown that this drawback can be overcome by changing the key in the watermarking system dependant on features of the incoming signal.

1. INTRODUCTION

A secret key watermarking system comprises a watermark embedder ***E*** (which embeds the payload P_I into an information signal X_I, generating Y_I) and a watermark detector ***D*** (which retrieves the payload P_I from the information signal Y_I). The secret key S_I, or the parameters required to generate that key, are locally stored in the embedder and in the detector in a secure way.

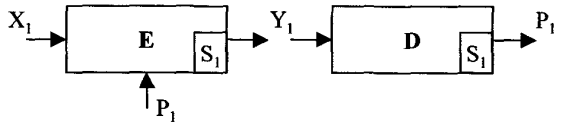

Fig. 1: Secret key watermarking system

As can be seen in Fig. 1, this representation is very much similar to a symmetric key cryptographic system. The secrecy of the system fully relies within the secret key (Kerkhoff's principle) and not within the watermark algorithm which may be public. Although watermarking is certainly on a lower level of security than cryptography (encryption is not bound by any considerations on content degradation, whereas watermarking is very much so) the "open source model" is to be preferred over "security by obscurity".
It should be noted that in most implementations of the system depicted in Fig. 1, the watermark embedder should contain a built-in watermark detector which prevents the embedding of a particular payload into content that has already been watermarked, using the same secret key. Due to the fact that for many secret key watermarking algorithms the embedding of different payloads results in a watermarked signal of which the watermark detector can not determine the payload.

2. DESCRIPTION OF THE PROBLEM

Many watermarking applications are characterised by the following properties:

- The watermark embedding is performed by content owners.
- The different entities that perform the embedding should not interfere with each other.
- Each watermark detector should be able to detect the payload embedded by any of the embedders.

Typical examples include copy protection [1] and broadcast monitoring [2], which is depicted in Fig. 2. A number of content owners are each supplied with a watermark embedder, such that they can embed a unique identifier (possibly supplied by a third party) into the content they generate and for which they own the rights.

0-7803-6297-7/00/$10.00

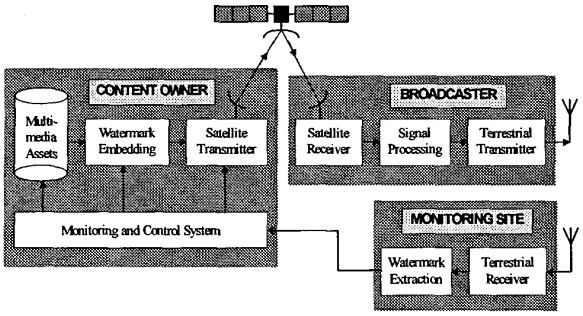

Fig. 2: Broadcast Monitoring

These content owners deliver the content to a number of broadcasters, who have to pay in order to have access to this content. Using a world wide network of monitoring stations equipped with watermark detectors it is possible to trace the usage of the content. Further, using appropriate database technology illegal usage is reported back to the content owner.

A straightforward approach to implement a watermark system would be to use the watermarking system as depicted in Fig. 1. One secret key S_1 is selected for all watermark embedders and detectors in the entire system. The payload P_1 consists of two parts:

- Part I: identifying the embedder,
- Part II: identifying the content.

However a hacker can render the payload detection of the watermarked signal impossible as depicted in Fig. 2. By feeding a first signal R_1 to a watermark embedder and by comparing (subtracting) the output and the input from the embedder. A difference signal can be created which can be used to make the detection of a second watermarked signal Y_1 impossible. It is sufficient to add (or subtract) the obtained difference signal to (from) the watermarked second signal Y_1. This operation will cause minor perceptual degradations to that signal, as the amplitude of the difference signal will be small. For most watermark schemes, this will render detection of the watermark payload impossible. An underlying assumption is that both signals R_1 and X_1 have been watermarked by the same algorithm, the same secret key S_1 and different payloads P_R and P_1.

A second approach would be to use a different secret key in each embedder. It is clear that, using the above described technique, a hacker can only compromise these signals which have been watermarked using his own secret key. However, increasing the number of secret keys in the system increases the implementation complexity of the detector which simultaneously has to search for watermarks trying all the possible secret keys.

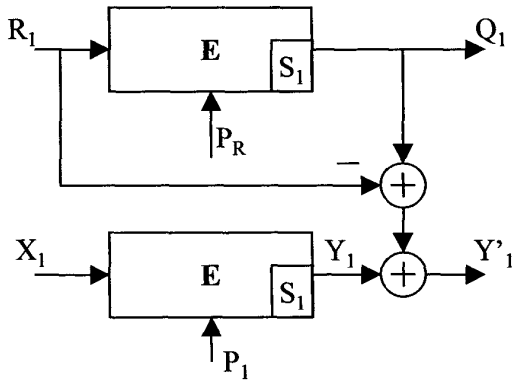

Fig. 3: Typical hack

For most watermark algorithms, the implementation complexity of the detector will increase linearly with the total number of secret keys it has to be able to accommodate.

3. PROPOSED SOLUTION

In this paper we propose a system which makes the hack as described in Fig. 3 impossible and which still ensures that the detector only has to search for watermarks corresponding to one particular secret key at a time. The basic idea is that the system only uses one secret key at a time, but that this secret key is not constant over time. Of course the watermark detector should somehow know which secret key has been used during the embedding. The idea is to make use of robust features of the signal to enable this. These robust features are combined into a robust signature. This signature then corresponds via a mapping to one of the secret keys S. The larger the number of bits in the robust signature and the number of "orthogonal" secret key patterns, the more difficult it will be for a hacker to compromise the system.
When the robust signature consists of many bits, only the mapping functions and the secret patterns need to be kept secret. The feature extraction and how these features are combined into a robust signature can be made public.

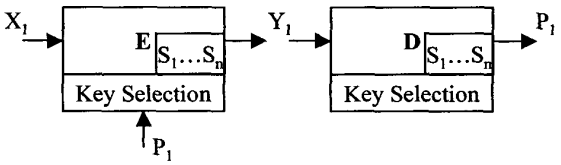

Fig. 4: Proposed solution

The features used, should be at least as robust as the secret key watermarking algorithm under consideration. To reduce complexity many video watermark accumulate

a number of video frames before doing a watermark detection. Thus in order not to increase the complexity of the watermark detection by a large factor it is necessary that the robust signature, and thus the secret key, does change slowly over time. Of course, the embedder and detector should be able to store the secret key patterns. In order to reduce the memory requirements the secret key patterns can be generated on the fly.

By letting features determine the secret key, the embedded watermark is totally dependent on the video. There have been published watermark schemes where the watermark also depends on the video [3][4], but in an essential different way. Their focus is more on invisibility then on security. These schemes locally scale the watermark, depending on the activity in that region of the frame. A region with a high activity (textured area's) will be scaled by a large factor, whilst flat area's will be scaled by a small factor.

4. EXAMPLE

To illustrate the above mentioned solution, let us consider the Philips secret key video watermarking system "JAWS" [4]. In this system, the secret key corresponds to a particular noise pattern or (depending on the payload) a set of noise patterns. A watermark is embedded by adding (or subtracting) cyclic shifted versions of these noise patterns to (or from) the video. Whether a noise pattern is added or subtracted and the vector over which it is cyclicly shifted depends on the payload [5]. Detection is based on correlation with the same noise patterns over all possible cyclic shifts. The payload is then retrieved by finding the shifts and the signs of the correlation peaks.

If the hack described in Fig. 3 is applied to this system the watermark detector will find multiple correlation peaks and therefore a possible wrong payload.

In order to apply the above mentioned solution, a number n of secret keys $S_1, S_2, \ldots, S_n$ have to be selected, corresponding to (sets of) noise patterns $P_1, P_2, \ldots, P_n$. Depending on the robust signature of the incoming video, for each video frame (or number of consecutive video frames) one particular (set of) noise pattern(s) is selected. A simple robust feature can be constructed by comparing the average luminance in a region of a frame with the average luminance of the total frame. A feature corresponds to a 1-bit if the average luminance in the region is larger then the one of the total frame. Otherwise the feature corresponds to a 0-bit. By subdividing the frame in m regions an m bit robust signature can be constructed. A mapping of 2^m to n selects the particular noise pattern. The watermark detector performs the same operations as to find out which particular noise pattern should be used for correlation. It is clear that the above described feature is robust to many signal processing operations.

5. CONCLUSIONS

In this paper we have presented a solution that applies to many digital watermarking applications where each watermark detector should be able to decode information that was inserted by different watermark embedders. The solution does not suffer from the simple attack, by adding a watermark with a different payload that was generated by embedding another video stream. The implementation complexity of the detector remains acceptable, even for a large number of watermark embedders. The basic idea is the use of a secret key which changes over time, depending on a robust signature of the input signal. The robust signature extraction algorithm and the watermark algorithm can be made public when the mapping function from robust signature to secret key and the secret keys are kept secret.

References:

[1] Jeffrey Bloom, Ingemar J. Cox, Ton Kalker, Jean-Paul Linnartz, Matt L. Miller, and C.B.S. Traw, "Copy protection for DVD video," Proceedings of the IEEE, Special Issue on Identification & Protection of Multimedia Information, vol. 87, no.7, pp. 1267 – 1276, July 1999.

[2] G. Depovere et al, "Digital watermarking for broadcast monitoring: The VIVA project", ICIP 99, paper 26AP1.1, Kobe, Japan, October 1999.

[3] Ingemar J. Cox, Joe Kilian, Tom Leighton and Talal Shamoon, "A Secure, Robust Watermark for Multimedia", Workshop on Information Hiding, Univ. of Cambridge, May 1996.

[4] T. Kalker, G. Depovere, J. Haitsma and M. Maes, "A Video Watermarking System for Broadcast Monitoring", proceedings of IS&T/SPIE/, security and watermarking of multimedia contents (EI25), San Jose, 1999.

[5] M. Maes, T. Kalker, J. Haitsma and G. Depovere, "Exploiting Shift Invariance to obtain a High Payload in Digital Image Watermarking", ICMS '99, Florence, Italy