A Cryptographic Method for Secure Watermark Detection

Michael Malkin[1] and Ton Kalker[2]

[1] Stanford University,
Stanford, CA, USA
mikeym@cs.stanford.edu
[2] Hewlett-Packard Laboratories
Palo Alto, CA, USA
ton.kalker@hp.com

Abstract. We present a semi-public key implementation of quantization index modulation (QIM) watermarking called Secure QIM (SQIM). Given a signal, a watermark detector can learn the presence of an SQIM watermark without learning anything anything else from the detection process. The watermark detector first transforms the signal with a secret transform, unknown to the detector, and then quantizes the transform coefficients with secret quantizers, also unknown to the detector. This is done with the use of homomorphic cryptosystems, where calculations are performed in an encrypted domain. A low-power, trusted, secure module is used at the end of the process and reveals only if the signal was watermarked or not. Even after repeated watermark detections, no more information is revealed than the watermarked status of the signals. The methods we present are for watermark systems with quantizers of stepsize 2.

1 Introduction

When watermarking occurs for the purposes of digital rights management (DRM), watermark embedding is performed in a trusted environment, while watermark detection is performed "in the wild". That is, the watermark detector is assumed to be a trusted party, but it is generally operating in a hostile environment where the end-user would like to circumvent the DRM. One way to keep the watermarking secret and functionality out of the hands of hostile parties is to embed it in a physically secure device, such as a smartcard, which can be operated in a black-box manner. However, such devices generally have very low computing capacity, and will be unable to perform watermark detection very quickly on their own. Our strategy, also proposed in other papers, is to have the watermark detector work in an encrypted domain and use a trusted secure device, called the *secure module*, to finish the detection process.

Secure QIM (SQIM) uses public and private keys much like public key cryptosystems such as RSA. The private key is used by the watermark embedder to generate watermarks while the public key is used by the watermark detector to

J. Camenisch et al. (Eds.): IH 2006, LNCS 4437, pp. 26–41, 2007.

perform watermark detection in an encrypted domain. Finally, the secure module uses the private key to decrypt the results produced by the watermark detector. The secure module must be initialized by communication with the watermark embedder to receive the private key information. In a sense, this system is not truly asymmetric but rather semi-asymmetric, since the aid of a trusted third party (the secure module) is required.

Our first goal is to ensure that the act of detecting a watermark reveals as little information as possible to the watermark detector. Because the secure module is low-power and low-bandwidth, our second goal is to ensure that the watermark detector takes on as much of the computational burden as is possible, and transmits a few bits to the secure module as possible. Two cryptosystems are used to allow the watermark detector to perform the necessary calculations without learning any information about the watermarking secrets. These systems are *homomorphic*, meaning that an operation performed on ciphertexts corresponds to another operation performed on plaintexts. For example, in the Paillier cryptosystem (see Section 3.1), if $E(\cdot)$ is the encryption function, then $E(x)E(y) = E(x + y)$. The homomorphic properties of these cryptosystems are what make it possible for the watermark detector to run the algorithm without learning anything.

However, even though the watermark detector gains no extra knowledge through the detection process, knowledge of the presence or absence of watermarks is sufficient to mount *oracle attacks* (see Cox and Linnartz [5], Venturini [18], and Li and Chang [14], for example). The purpose of these attacks is to find the boundary separating watermarked signals from non-watermarked signals, and use this boundary to learn the watermarking secret. Such attacks are much more powerful than attacks on the cryptosystems presented in this paper, and are possible whenever a watermark detector can test signals for watermarks.

One defense against oracle attacks is to increase the time required for watermark detection, effectively limiting the speed of the "oracle" (See Venturini [18]). This would not be possible with a fully assymetric watermarking scheme, since the speed of such a watermark detector would be limited only by the speed of the machine that is running it. A trusted secure module could have a built-in delay, or a limit on the number of watermark detections per minute, and could thereby help to slow the rate of convergence of oracle attacks.

On the other hand, the use of a secure module introduces side channel attacks, for example timing attacks (see Kocher [10], and Brumley and Boneh [2]), and power attacks (see Kocher et al. [12]). In these attacks, the secure module is monitored externally to guess at the operations occurring internally. An implementation of SQIM would have to take side channel attacks into account, but a detailed discussion of these attacks is beyond the scope of this paper.

Quantization Index Modulation (QIM), developed by Chen and Wornell [4], embeds a watermark into an signal by manipulating the signal so that transform coefficients are quantized in a specific manner. A watermark detector transforms a signal and checks to see if the transform coefficients are appropriately quantized. There are two phases to securely detecting a QIM watermark. First