Content-Aware Steganography: About Lazy Prisoners and Narrow-Minded Wardens

Richard Bergmair[1] and Stefan Katzenbeisser[2]

[1] Computer Laboratory, University of Cambridge
[2] Institut für Informatik, Technische Universität München
{rbergmair,skatzenbeisser}@acm.org

Abstract. We introduce content-aware steganography as a new paradigm. As opposed to classic steganographic algorithms that only embed information in the syntactic representation of a datagram, content-aware steganography embeds secrets in the semantic interpretation which a human assigns to a datagram. In this paper, we outline two constructions for content-aware stegosystems, which employ, as a new kind of security primitive, problems that are easy for humans to solve, but difficult to automate. Such problems have been successfully used in the past to construct Human Interactive Proofs (HIPs), protocols capable of automatically distinguishing whether a communication partner is a human or a machine.

1 Content-Aware Steganography

In his 1984 landmark paper [23], Gustavus Simmons illustrated what is now widely known as the *prisoners' problem*: Two accomplices in a crime, Alice and Bob, are arrested in separate cells. They want to coordinate an escape plan, but their only means of communication is by way of messages conveyed for them by Wendy the warden. Should Alice and Bob try to exchange messages that are not completely open to Wendy, or ones that seem suspicious to her, they will be put into a high security prison no one has ever escaped from. Simmons' solution to the prisoners' problem is phrased in an interesting way: Alice and Bob "will have to deceive the warden by finding a way of communicating secretly in the exchanges, i.e. of establishing a 'subliminal channel' between them in full view of the warden, even though the messages themselves contain no secret (to the warden) information" [23]. In other words, Alice is trying to convey a particular piece of information which is represented as a single *datagram*. This datagram is available to both Wendy and Bob—but it contains different *information* to Wendy than to Bob.

Informally speaking, a subliminal channel is one that transmits datagrams that have at least two possible interpretations. Each datagram is intentionally given an obvious interpretation (the cover) that is innocuous to Wendy, and a non-obvious interpretation (the secret) that is suspicious to Wendy, and thus cannot be transmitted in plain sight. The security of the stegosystem usually relies on some assumption of an advantage that Bob has over Wendy, when it comes

J. Camenisch et al. (Eds.): IH 2006, LNCS 4437, pp. 109–123, 2007.

to the *interpretation* of the message: Bob can interpret the message with regard to its secret meaning, while Wendy can only interpret the message as the cover.

In the past, many stegosystems have been constructed, most of them using images, digital audio, or video as cover. Consider for example a simplistic LSB scheme for image-based steganography in which the cleartext message is written into the LSBs of an image without any further cryptographic concealment. The datagram has an obvious interpretation, which is visual perception by a human user of the pattern that appears on screen when it is opened in their favourite image viewer. It also has a non-obvious interpretation, which is to extract the LSBs and view their concatenation, say, in a hex-editor. Under the assumption that Alice constantly sends Bob bitmap images that Wendy is not willing to wade through with a hex-editor, this simplistic system might be attributed some kind of security. However, Wendy will probably try to *automatically* analyze all datagrams exchanged between Alice and Bob to gain knowledge of a subliminal channel. This notion of automaticity in steganalysis has probably received too little attention in the past, which is why we shall, in this paper, take the challenging point of view that a stego object should not be considered perfectly secure as long as its *semantics* are prone to *automatic interpretation by a machine*.

Due to recent progress in the field of steganalysis (see for example [17]), LSB substitution techniques must be considered completely insecure today. To understand why LSB steganography was compromised, it is important to bear in mind that a bitmap image is not just a sequence of bytes, but rather a representation for some specific semantic content. It could, for example, be a vector drawing consisting of uniformly colored geometric shapes. If a set of pixels can be identified as representing, say, an oval shape colored in a certain tone of blue, and half of these pixels deviate in their color by the LSB, this might give us some evidence of steganography taking place. A 24-bit bitmap might also be a photograph taken by a digital camera with a CCD that leaves noise with special characteristics in the images [20]. If these characteristics cannot be found in the LSBs of the image, then again we have gained evidence to suspect that steganography is taking place.

We believe the way in which LSB substitution has been compromised is stereotypical for how the steganography vs. steganalysis battle is usually fought, namely by steganalysis exploiting the false assumption made by steganography that a meaningful digital object can be specified *solely* in terms of syntactic properties. Stegosystems are usually broken by exploiting *semantic* inconsistencies introduced into the cover when hiding a secret. This is a limitation which is inherent with every steganographic system that takes a cover and applies modifications in order to obtain a stego object: an attacker that possesses a more accurate semantic cover model than the embedder can break the system easily. Thus, a security vulnerability is necessarily opened in any steganographic system whose participants are computers that employ state-of-the art cover models, as soon as the state of the art improves.

In this paper, we propose an alternative view of steganography, which takes semantic aspects into account and hides information in the *semantics* (rather