On the Limits of Steganography

Ross J. Anderson and Fabien A. P. Petitcolas

***Abstract*—In this paper, we clarify what steganography is and what it can do. We contrast it with the related disciplines of cryptography and traffic security, present a unified terminology agreed at the first international workshop on the subject, and outline a number of approaches—many of them developed to hide encrypted copyright marks or serial numbers in digital audio or video. We then present a number of attacks, some new, on such information hiding schemes. This leads to a discussion of the formidable obstacles that lie in the way of a general theory of information hiding systems (in the sense that Shannon gave us a general theory of secrecy systems). However, theoretical considerations lead to ideas of practical value, such as the use of parity checks to amplify covertness and provide public key steganography. Finally, we show that public key information hiding systems exist, and are not necessarily constrained to the case where the warden is passive.**

***Index Terms*—Copyright protection, cryptography, data compression, image registration, jitter, motion pictures, multimedia systems, music, observability, pseudonoise coded communication, redundancy, spread spectrum communication, software protection.**

I. Introduction

While classical cryptography is about concealing the content of messages, steganography is about concealing their existence. It goes back to antiquity: Herodotus relates how the Greeks received warning of Xerxes' hostile intentions from a message underneath the wax of a writing tablet, and describes a trick of dotting successive letters in a covertext with secret ink, due to Aeneas the Tactician. Kahn tells of a classical Chinese practice of embedding a code ideogram at a prearranged place in a dispatch; the same idea arose in medieval Europe with grille systems, in which a paper or wooden template would be placed over a seemingly innocuous text, highlighting an embedded secret message.

Such systems only make sense when there is an opponent. This opponent may be passive, and merely observe the traffic, or he may be active and modify it. A famous case dates back to 1586, when Mary Queen of Scots was conspiring to have Queen Elizabeth of England assassinated, with a view to taking over the English throne. However, the cipher she used was broken, and the English secret police obtained the would-be assassins' names by forging a postscript to a letter she wrote to the chief conspirator, asking for "the names and qualities of the six gentlemen which are to accomplish the designment." This led to their arrest and execution, as indeed to Mary's the following year. In this century, postal censors have deleted lovers' X's from letters, shifted the hands of watches in shipments, and even rephrased telegrams; in one case, a censor changed "father is dead" to "father is deceased," which elicited the reply "is father dead or deceased?" [23].

The study of this subject in the scientific literature may be traced to Simmons, who in 1983 formulated it as the "Prisoners' Problem" [43]. In this scenario, Alice and Bob are in jail, and wish to hatch an escape plan; all their communications pass through the warden, Willie, and if Willie detects any encrypted messages, he will frustrate their plan by throwing them into solitary confinement. So they must find some way of hiding their ciphertext in an innocuous looking covertext. As in the related field of cryptography, we assume that the mechanism in use is known to the warden, and so the security must depend solely on a secret key that Alice and Bob have somehow managed to share.

There are many real life applications of steganography. Apparently, during the 1980's, Margaret Thatcher became so irritated at press leaks of cabinet documents that she had the word processors programmed to encode their identity in the word spacing, so that disloyal ministers could be traced. Similar techniques are now undergoing trials in an electronic publishing project, with a view to hiding copyright messages and serial numbers in documents [30].

Simmons' formulation of the Prisoners' Problem was itself an instance of information hiding. It was a ruse to get the academic community to pay attention to a number of issues that had arisen in a critical but at that time classified application—the verification of nuclear arms control treaties. The United States and the USSR wanted to place sensors in each others' nuclear facilities that would transmit certain information (such as the number of missiles) but not reveal other kinds of information (such as their location). This forced a careful study of the ways in which one country's equipment might smuggle forbidden data past the other country's monitoring facilities [44], [46].

Steganography must not be confused with cryptography, where we transform the message so as to make its meaning obscure to a person who intercepts it. Such protection is often not enough. The detection of enciphered message traffic between a soldier and a hostile government, or between a known drug smuggler and someone not yet under suspicion has obvious implications; and recently, a U.K. police force concerned about criminal monitoring of police radios has discovered that it is not enough to simply encipher the traffic, as criminals detect, and react to, the presence of encrypted communications nearby [49].

Manuscript received March 1997; revised August 1997. The work of F. A. P. Petitcolas was supported by the Intel Corporation under the Grant "Robustness of Information Hiding Systems."

The authors are with the Computer Laboratory, University of Cambridge, Cambridge CB2 3QG, U.K. (e-mail: rja14@cl.cam.ac.uk; fapp2@cl.cam.ac.uk).

Publisher Item Identifier S 0733-8716(98)01112-3.

In some applications, it is enough to hide the identity of either the sender or the recipient of the message, rather than its very existence. Criminals often find it sufficient for the initiator of a telephone call to be anonymous. Indeed, the main practical problem facing law enforcement and intelligence agencies is "traffic selection"—deciding which calls to intercept—and because of the huge volume of traffic, this must usually be done in real time [28].

The techniques criminals use to thwart law enforcement vary from country to country. United States villains use "tumblers"—cellular phones that continually change their identity, using genuine identities that have either been guessed or intercepted. In France, drug dealers drive around with a cordless phone handset until a dial tone is found, then stop to make a call [26]; while in one U.K. case, a drug dealer physically tapped into a neighbor's phone [13]. All these techniques also involve theft of service, whether from the phone company or from one of its customers; so this is one field where the customer's interest in strong authentication and the police interest in signals intelligence coincide. However, authentication itself is not a panacea. The introduction of GSM, with its strong authentication mechanisms, has led crooks to buy GSM mobile phones using stolen credit cards, use them for a few weeks, and then dispose of them [53].

Military organizations also use unobtrusive communications. Their preferred mechanisms include spread spectrum and meteor scatter radio [39], which can give various combinations of resistance to detection, direction finding, and jamming. They are vital for battlefield communications, where radio operators who are located are at risk of being attacked. On the Internet, anonymous remailers can be used to hide the origin of an e-mail message, and analogous services are being developed for other protocols such as `ftp` and `http`[1] [16], [38].

Techniques for concealing meta-information about a message, such as its existence, duration, sender and receivers, are collectively known as traffic security. Steganography is often considered to be a proper subset of this discipline rather than being co-extensive with it, so we shall now try to tie down a definition.

II. What Is Steganography?

Classical steganography concerns itself with ways of embedding a secret message (which might be a copyright mark, a covert communication, or a serial number) in a cover message (such as a video film, an audio recording, or computer code). The embedding is typically parameterized by a key; without knowledge of this key (or a related one) it is difficult for a third party to detect or remove the embedded material. Once the cover object has material embedded in it, it is called a stego object. Thus, for example, we might embed a mark in a covertext giving a stegotext, or embed a text in a cover image giving a stego-image; and so on. (This terminology was agreed at the First International Workshop on Information Hiding [35].)

There has been a rapid growth of interest in this subject over the last two years, for two main reasons. First, the publishing and broadcasting industries have become interested in techniques for hiding encrypted copyright marks and serial numbers in digital films, audio recordings, books, and multimedia products; an appreciation of new market opportunities created by digital distribution is coupled with a fear that digital works could be too easy to copy. Secondly, moves by various governments to restrict the availability of encryption services have motivated people to study methods by which private messages can be embedded in seemingly innocuous cover messages. The ease with which this can be done may be an argument against imposing restrictions [15].

Other applications for steganography include the automatic monitoring of radio advertisements, where it would be convenient to have an automated system to verify that ads are played as contracted; indexing of videomail, where we may want to embed comments in the content; and medical safety, where current image formats such as DICOM separate image data from the text (such as the patient's name, date, and physician), with the result that the link between image and patient occasionally gets mangled by protocol converters. Thus embedding the patient's name in the image could be a useful safety measure.

Where the application involves the protection of intellectual property, we may distinguish between watermarking and fingerprinting. In the former, all the instances of an object are marked in the same way, and the object of the exercise is either to signal that an object should not be copied, or to prove ownership in a later dispute. One may think of a watermark as one or more copyright marks that are hidden in the content.

With fingerprinting, on the other hand, separate marks are embedded in the copies of the object that are supplied to different customers. The effect is somewhat like a hidden serial number: it enables the intellectual property owner to identify customers who break their license agreement by supplying the property to third parties. In one system we developed, a specially designed cipher enables an intellectual property owner to encrypt a film soundtrack or audio recording for broadcast and issue each of his subscribers a slightly different key; these slight variations cause imperceptible errors in the audio decrypted using that key, and the errors identify the customer. The system also has the property that more than four customers have to collude in order to completely remove all the evidence identifying them from either the keys in their possession or the audio that they decrypt [6].

Using such a system, a subscriber to a music channel who posted audio tracks to the Internet, or who published his personal decryption key there, could be rapidly identified. The content owner could then either prosecute him, revoke his key, or both.

There is a significant difference between classical steganography, as modeled in the Prisoners' Problem, and copyright marking. In the former, a successful attack consists of the warden observing that a given object is marked. In the second, all the participants in the scheme may be aware that marks are in use—so some effects of the marks may be observable (marks should remain below the perceptual threshold, but they may alter the content's statistics in easily measurable ways). So a successful attack does not mean detecting a mark, but

[1] Found at www.anonymizer.com.

rendering it useless. This could be done by removing it, or by adding many more marks to prevent a court from telling which one was genuine. Blocking such attacks may involve embedding a signature by the customer in the content [36] or involving a public timestamping service in the marking process.

III. The State of the Art

Prudent cryptographic practice assumes that the method used to encipher data is known to the opponent, and that security must lie in the choice of key. This principle was first enunciated by Kerckhoffs in 1883 [24], and has been since borne out by long and hard experience [23]. It should be an obvious requirement for protection mechanisms designed to provide evidence, as one can expect that them to be scrutinized by hostile expert witnesses in open court [2].

So one might expect that designers of copyright-marking systems would publish the mechanisms they use and rely on the secrecy of the keys employed. Sadly, this is not the case; many purveyors of such systems keep their mechanisms subject to nondisclosure agreements, sometimes offering the rationale that a patent is pending. So we will briefly survey a few systems that have been described in public, or of which we have information.

A. Simple Systems

A number of computer programs are available that will embed information in an image. Some of them just set the least significant bits of the image pixels to the bits of the embedded information [51]. Information embedded in this way may be invisible to the human eye [27] but is trivial for an alert third party to detect and remove.

Slightly better systems assume that both sender and receiver share a secret key and use a conventional cryptographic keystream generator [40] to expand this into a long pseudorandom keystream. The keystream is then used to select pixels or sound samples in which the bits of the ciphertext are embedded [15].

Not every pixel may be suitable for encoding ciphertext: changes to pixels in large fields of monochrome color, or that lie on sharply defined boundaries, might be visible. Therefore some systems have an algorithm that determines whether a candidate pixel can be used by checking that the variance in luminosity of the surrounding pixels is neither very high (as on a boundary) nor very low (as in a monochrome field). Wherever a pixel passes this test, we can tweak its least significant bit to embed a bit of our message.

Such schemes can be destroyed in a number of ways by an opponent who can modify the stego-image. For example, almost any trivial filtering process will change the value of many of the least significant bits. One possible countermeasure is to use redundancy: either apply an error correcting code, or simply embed the mark a large number of times. For example, the "Patchwork" algorithm of Bender *et al.* hides a bit of data in an image by increasing the variance in luminosity of a large number of pseudorandomly chosen pixel pairs [9]; and a similar system was proposed by Pitas [37]. Much the same techniques can be used to mark digital audio as well.

One way in which we have attacked such systems is to break up the synchronization needed to locate the samples in which the mark is hidden: pictures, for example, can be cropped. In the case of audio, we have developed a simple but effective desynchronization attack—we randomly delete a small proportion of sound samples, and duplicate a similar number of others. This introduces a jitter of a few tens of milliseconds, that is tiny compared to the precision with which the original sounds were in most cases generated, but is sufficient to confuse a typical marking scheme.

With a pure tone, we can delete or duplicate one sample in 8000, and with classical music we can delete or duplicate one sample in 500, without the results being perceptible either to us or to laboratory colleagues. Using more sophisticated resampling and filtering algorithms, we can obtain a 1 in 500 jitter in pure tone, and 1 in 50 in speech, without making a perceptible difference. (The result for classical music can also be improved significantly, but the precise figure depends on the music.)

B. Operating in a Transform Space

A systematic problem with the kind of scheme described above is that those bits in which one can safely embed covert data are by definition redundant—the attacker will be unaware that they have been altered—and it follows that they might be removed by an efficient compression scheme. The interaction between compression and steganography is a recurring thread in the literature.

When we know in advance what compression scheme will be used, we can often tailor an embedding method to get a quite reasonable result. For example, with `.gif` files, one can swap colors for similar colors (those that are adjacent in the current palette) [22], while if we want to embed a message in a file that may be subjected to JPEG compression and filtering, we can embed it in multiple locations [25], [29] or, better still, embed it in the frequency domain by altering components of the image's discrete cosine transform. A particularly detailed description of such a technique may be found in [12]; this technique, being additive, has the property that if several marks are introduced in succession, they can all be detected (thus it is prudent for the originator of the content to use a digital timestamping service[2] in conjunction with the marking system, so that the priority of the genuine mark can be established). Other schemes of this kind include, for example, [10] and [25].

Such "spread spectrum" techniques are often tuned to the characteristics of the cover material. For example, one system marks audio in a way that exploits the masking properties of the human auditory system [11].

Masking is a phenomenon in which one sound interferes with our perception of another sound. Frequency masking occurs when two tones that are close in frequency are played at the same time. The louder tone will mask the quieter [20], [34]. However this does not occur when the tones are far apart in frequency. It has also been found that when a pure tone

[2] Found at www.surety.com.

is masked by wide-band noise, only a small band centered about the tone contributes to the masking effect [31]. Similarly, temporal masking occurs when a low-level signal is played immediately before or after a stronger one. For instance after we hear a loud sound, it takes a little while before we can hear a quiet one.

MPEG audio compression techniques exploit these characteristics [1], but it remains possible to exploit them further by inserting marks that are just above the truncation threshold of MPEG but still below the threshold of perception [11]. In general, a copyright mark's existence may be detected by statistical tests while it remains undetectable by humans; the real question is whether it can be damaged beyond later recognition without introducing perceptible distortion.

Embedding data in transformed content is not restricted to the "obvious" transforms that are widely used for compression, such as discrete cosine, wavelet, and fractal transforms. A recent interesting example has been suggested in [17]: this "echo hiding" technique marks audio signals by adding an echo. This echo might have a delay of 0.5 ms to signal a "0" and 1.0 ms to signal a "1;" these delays are too short to be perceptible in most circumstances, but can be detected using cepstral transforms.

C. A General Model

The general model of steganography we have developed in the above sections is that Alice embeds information by first applying a transform to the covertext, and then tweaking a subset of the bits of the transformed object that are now redundant. In this context, redundant means that a nontrivial subset of them, that is selected randomly to be of a given size, can have their values altered without being detected easily or at all by an opponent who does not know which subset to examine.

We will not expect to find high bandwidth channels, as these would correspond to redundancy that could economically be removed. However, the design of compression schemes is limited in most cases by economic factors; the amount of computation that we are prepared to do in order to replace a certain amount of communication is not infinite. If we are prepared to do a little more work than the "normal" user of the system, we will be able to exploit a number of low-bandwidth stego channels.

However, the warden may be willing to do even more work, and the apparent redundancy that we exploit will fall within his ability to model. This may be especially so if the warden is a person with access to future technology—for example, a pirate seeking to remove the watermark or fingerprint embedded in a 1997 music recording using the technology available in 2047. This is a serious concern with copyright, that may subsist for a long time (typically 70 years after the author's death for text and 50 years for audio). Even where we are concerned only with the immediate future, the industry experience is that it is a "wrong idea that high technology serves as a barrier to piracy or copyright theft; one should never underestimate the technical capability of copyright thieves [18]." Such experience is emphasized by the recent success of criminals in cloning the smartcards used to control access to satellite TV systems [5].

When such concerns arise in cryptography—for example, protecting traffic that might identify an agent living under deep cover in a foreign country—the standard solution is to use a one-time pad; Shannon provided us with a proof that such systems are secure regardless of the computational power of the opponent [42]. Simmons provided us with a comparable theory of authentication that has been applied in nuclear weapons command and control [45]. Yet we still have no comparable theory of steganography.

In the next section, we will discuss the formidable obstacles to such a theory, and indicate how some theoretical ideas have nonetheless led to useful improvements in the state of the art.

IV. Theoretical Limits

Can we get a scheme that gives unconditional covertness, in the sense that the one-time pad provides unconditional secrecy?

Suppose that Alice uses an uncompressed digital video signal as the covertext, and then encodes ciphertext at a very low rate. For example, the kth bit of ciphertext becomes the least significant bit of one of the pixels of the kth frame of video, with the choice of pixel being specified by the kth word of a shared one time pad. Then we intuitively expect that attacks will be impossible: the ciphertext will be completely swamped in the covertext's intrinsic noise. Is there any way this intuitive result could be proved?

We must first ask what a proof of covertness would look like. A working definition of a secure stegosystem might be one for which the warden cannot differentiate between raw covertext and the stegotext containing embedded information, unless he has knowledge of the key. As with cryptography, we might take the warden to be a probabilistic polynomial Turing machine in the case where we require computational security, and assume that he can examine all possible keys in the case where we require unconditional security.

In the latter case, he will see the actual embedded message, so the system must generate enough plausible embedded messages from any given stegotext, and the number of such messages must not vary in any usable way between the stegotext and a wholly innocent covertext.

This much is straightforward, but what makes the case of steganography more difficult than secrecy or authenticity is that we are critically dependent on our model of the covertext.

A. What If Perfect Compression Existed?

Workers in information theory often assume that any information source can be compressed until there is no redundancy left. This assumption may be very useful in proving asymptotic bounds and capacity results, but has a rather curious effect when applied to steganography.

Suppose that such a perfect coding scheme was actually instantiated in a physical black box that could both compress and decompress data of a particular type (audio, video, whatever). Completely efficient compression means that the compressed objects would be dense in the set of bit strings of the same length. Thus Alice could take an arbitrary ciphertext message that she wants to hide and run it through the decompressor. The result would be an acceptable audio recording, video film or whatever.

The above is not a rigorous proof. It is conceivable, for example, that a device might decompress a random bit string of length n to a particular type of object with a probability polynomial in $1/n$. This would suffice for many information theoretic results to go over, while invalidating the above argument. Nonetheless, it indicates that many classical intuitions of information theory serve us poorly when dealing with steganographic systems. It points to some interesting research problems in closing the gap between the two, and tells us that practical steganography is only an issue where compression is inefficient. Where efficient compression is available, information hiding will usually be either trivial or impossible, depending on the context.

B. Entropy

Entropy arguments are used in conventional information theory; how far will they get us in steganography?

Assuming that the material to be embedded is indistinguishable from random data (as would be the case were it competently encrypted), then entropy will be strictly additive: the entropy of the stegotext S will equal the entropy of the covertext C plus the entropy of the embedded material E

$$H(S) = H(C) + H(E). \qquad (1)$$

Thus in order to make our embedding process secure against an opponent who merely has to detect the presence or absence of embedded material, it appears that we have two alternatives.

1) Keep $H(E)$ much less than the uncertainty in the opponent's measurement of $H(C)$.
2) Find some way of processing C to reduce its entropy by an amount that can then be made up be adding E. For example, one might use a noise reduction or lossy compression algorithm to remove some unnecessary information from C before embedding E.

The problem is that we do not know how competent our opponent is at measuring the entropy of the covertext we are using, or, equivalently, at discriminating signal from noise. We will often be up against an opponent of unpredictable power (a pirate attacking our system a generation from now), and these are precisely the circumstances where we may want a security proof.

But the more stegotext we give the warden, the better he may be able to estimate the statistics of the underlying covertext, and so the smaller the rate at which Alice will be able to tweak bits safely. The rate might even tend to zero, as was noted in the context of covert channels in operating systems [32]. However, as a matter of empirical fact, channels do exist in which ciphertext can be inserted at a positive rate [15], and people have investigated correlations in various types of content such as digital video [47]. So measuring entropy may be useful in a number of applications.

But is there any prospect of developing steganographic techniques that we can prove will resist an opponent of arbitrary ability?

C. Selection Channel

Our next idea is inspired by the correction channel that Shannon uses to prove his second coding theorem. (This is the channel that someone who can see both the transmitted and received signals uses to tell the receiver which bits to tweak; it produces various noise and error correction bounds [41].)

In a similar way, when Alice and Bob use a shared one-time pad to decide which covertext bit will be marked with the next ciphertext bit, we can think of the pad as a selection channel. If Willie is computationally unbounded, he can try all possible pads (including the right one), so the number of them that yield a plausible ciphertext must be large enough that he cannot reasonably accuse Alice of sending stegotext rather than an innocent message.

It may be useful at this point to recall the book cipher. The sender and receiver share a book and encipher a message as a series of pointers to words. So the cipher group "78216" might mean page 78, paragraph 2, and the 16th word. Book codes can be secure provided that the attacker does not know which book is in use, and care is taken not to reuse a word (or a word close enough to it) [23]. The book cipher is a kind of selection channel. The model of computation may appear to be different, in that a book cipher starts off with the book and then generates the ciphertext, whereas in a stegosystem, we start off with the text to be embedded and then create the stegotext. However in the case where the selection channel is truly random (a one-time pad), they are the same, in that an arbitrary message can be embedded in an arbitrary covertext of sufficient length.

A repetitive book will have a lower capacity, as we will be able to use a smaller percentage of its words before correlation attacks from the context become possible. Similarly, if the covertext to be used in a stegosystem has unusual statistics (such as an unequal number of zeros and ones) then its stego capacity will be lower, as only a small proportion of candidate ciphertexts would look random enough.

D. The Power of Parity

We mentioned systems that generate a number of candidate locations for a ciphertext bit and then filter out the locations where actually embedding a bit would have a significant effect on the statistics thought to be relevant (in the case of hiding in an image, this could mean avoiding places where the local variance in luminosity is either very low or very high).

Our selection channel approach led us to suggest a better way [3]. We use our one-time pad (or keystream generator) to select not one pixel, but a set of them, and embed the ciphertext bit as their parity. This way, the information can be hidden by changing whichever of the pixels can be changed least obtrusively.

From the information-theoretic point of view, if each bit of the covertext is "1" with probability 0.6, then the probability that a bit pair will have parity 1 is 0.52; if we move to triples, the parity is 1 with probability 0.504, and so on. Thus by encoding each embedded bit as the parity of k bits of stegotext, we can reduce the effect that the embedding process has on the statistics of the stegotext below any arbitrary threshold; and as the improvement is geometric, we will not in practice have to increase k very much.

There is an interesting tradeoff: the more bits in the selection channel (i.e., the greater the value of k), the more bits we can hide in the covertext. In practice, our selection channel will be a cryptographic pseudorandom number generator, and we can draw from it as many bits as we like.

There are still limits. For example, suppose that there is an allowed set of cover texts M (we might be using the cover of a news agency; we have to report a reasonably truthful version of events, and transmit photographs—perhaps slightly doctored—of events that actually took place). Suppose also that there is an allowed set of encodings E, and that each hidden bit is embedded by a choice of an encoding rule (such as a parity check in the method described above). Then the covert capacity will be at most $H(E) - H(M)$. But this gives us an upper bound only; it does not give us useful information on how much information may safely be hidden.

E. Equivalence Classes

Suppose Alice uses a keyed cryptographic hash function to derive one bit from each sentence of a document. She may even have a macro in her word processor that checks every sentence as she finishes typing it and beeps if the output of the cryptographic hash function is not equal to the next bit of the message she wishes to embed. This alarm will go off about every other sentence, which she can then rewrite.

If she just uses standard synonym pairs such as [is able ↔ can], then clearly she must not alter their statistics to the point that Willie can detect the change. It is even an open question whether a computer can alter a natural language text in a way that is undetectable to a human [48]—that is, embed a ciphertext using the technique described above—and the problem is commended to the research community as the "Stego Turing Test." Conversely, writing a program to scan for human-inserted steganography might be rather hard. Recent work on natural language-based stego is described in [52].

The use of synonyms to encode embedded messages is a special case of using equivalence classes of messages; these can also arise naturally in other applications. For example, when making a map from a larger scale map, many arbitrary decisions have to be made about which details to incorporate, especially with features such as coastlines that are to some extent fractal [33]. Also, when software is written, it contains "birthmarks" such as the order in which registers are pushed and popped, and these were used by IBM in litigation against software pirates who had copied their PC-AT ROM [21].

Equivalence classes of messages are tied up with compression. If covertext C_1 has a meaning or effect that is equivalent to that of covertext C_2, then a compression algorithm need only select one representative from this equivalence class. However, if $C_1 \neq C_2$, then this choice throws away information, and the compression is lossy. Again, we get a bound on the stego channel capacity: it is the difference between lossy and lossless compression. Once more though, this is an upper bound rather than a safety bound, and is not much help against a powerful opponent.

It must be said that not all steganographic techniques involve equivalence classes. It is possible to create a series of images each of which differs only imperceptibly from the next, but such that the starting and final images are clearly different. This is relevant to the case where the warden is allowed to insert only so much distortion into messages; beyond a certain limit he might be held, in the absence of any hard evidence of covert activity by a prisoner, to have violated that prisoner's human rights.

A purist might conclude that the only circumstance in which Alice can be certain that Willie cannot detect her messages is when she uses a true subliminal channel ([7], and references therein). However, other interesting things can be said about steganography.

V. Active and Passive Wardens

We pointed out above that while an attack on a classical steganographic system consisted of correctly detecting the presence of embedded matter, an attack on a copyright marking scheme consists of rendering the mark useless.

There is a critical distinction between passive wardens, who monitor traffic and signal to some process outside the system if unauthorized traffic is detected, and active wardens, who try to remove all possible covert messages from traffic that passes through their hands. In classical systems, the wardens could be either active or passive; while in marking systems, we are usually concerned with active wardens such as software pirates.

Consider the marking of executable code. Software birthmarks, as mentioned above, have been used to prove the authorship of code in court. They were more or less "automatically" generated when system software was hand assembled, but they must be produced more deliberately now that most code is compiled. One technique is to deliberately mangle the object code: the automatic, random replacement of code fragments with equivalent ones is used by Intel to customize security code [8].

One can imagine a contest between software authors and pirates to see who can mangle code most thoroughly without effecting its performance too much. If the author has the better mangler, then some of the information he adds will be left untouched by the pirate; but if the pirate's code mangler is aware of all the equivalences exploited by the author's, he may be able to block the stego channel completely. In general, if an active warden's model of the communication is as good as the communicating parties' model, and the covertext information separates cleanly from the usable redundancy, then he can replace the latter with noise.

In many other cases, the stego channel is highly bound up with the covertext. There have been measurements of the noise that can be added to a `.gif` file before the image quality is perceptibly degraded [22], and of the noise that can imperceptibly be added to digitized speech [15].

The point here is that if Alice can add an extra X% of noise without effecting the picture, then so can Willie; but where the noise and the signal do not separate cleanly, both are limited to marking bits that are selected according to some rule or at random. Here, Alice can stop Willie from finding out which X% carries the covert message by using a keystream to select the bits she will mark. In this case, all Willie will be able to do is to cut the bandwidth of the channel by adding his own random marks—a scenario that has been explored in the context of covert channels in operating systems [50].

This bandwidth limitation will also be effective against systems that embed each ciphertext bit as a parity check of a number of stegotext bits. When the warden is active, the more covertext bits we use in each parity check, the more easily he will be able to inject noise into our stegotext.

VI. Public Key Steganography

Until recently, it was generally assumed that, in the presence of a capable motivated opponent, steganography required the pre-existence of a shared secret, so that the two communicating parties could decide which bits to tweak.

A. With a Passive Warden

In an early (workshop) version of this paper [3], we showed that public-key steganography is possible in the presence of a passive warden. Given a covertext in which any ciphertext at all can be embedded, there will usually be a certain rate at which its bits can be tweaked without the warden noticing. So suppose that Alice can modify at least one out of every k bits of the covertext. This means that Willie cannot distinguish the parity of each successive block of k bits from random noise, and it follows that Alice can encode an arbitrary pseudorandom string in these parities.

This pseudorandom material lies in plain sight; anyone can read it. So Willie cannot tell the difference between stegotext and pure covertext by randomness tests; a suitable parity check function will extract pseudorandom-looking data from any covertext in which information could have been embedded at all.

Now suppose that Alice and Bob did not have the opportunity to agree on a secret key before they were imprisoned, but that Bob has a public key that is known to Alice. She can take her covert message, encrypt it under his public key, and embed it as the parity of successive blocks. Each possible recipient will then simply try to decrypt every message he sees, and Bob alone will be successful. In practice, the value encrypted under a public key could be a control block consisting of a session key plus some padding, and the session key would drive a conventional steganographic scheme.

Normal public key cryptography means that users can communicate confidentially in the absence of previously shared secrets; our construction of public key steganography shows that they can also communicate covertly (if this is at all possible for people who already share a secret).

B. With an Active Warden

The open question left in [3] was whether public key steganography was possible in the presence of a warden who is active rather than passive. The original construction fails in this case, as Willie can also tweak one bit in every k; he could even set the parity of each successive block to zero. We will now indicate how concealed public key communication may still be possible in the presence of an active warden provided that the model of the Prisoners' Problem is changed slightly.

We will assume that the stegomessages Alice sends to Bob will be sent to other recipients too, such as a mailing list or usenet newsgroup. We will also assume that Willie and Alice are each able to tweak, at most, one bit in k of the content (as above, Willie might infringe Bob's rights if he distorts the communication channel to the point that it becomes unusable). Alice can choose a short one-time key that selects some permutation of the covertext bits, and she hides a message as the parity of successive k-tuples of bits in this permuted sequence. Willie, suspecting that this method may be in use, alters 1 in k of the stegotext bits; this is the best he can do because he does not know what one-time key Alice used. This corrupts most of the bits in Alice's message, but not all of them; asymptotically, about $1/e$ k-tuples will be unaffected, so there will be a positive residual channel capacity. Given a suitable error correcting code, Alice can still send a message encrypted using Bob's public key.

Once Bob has received the message, Alice broadcasts her short one-time key. Bob now applies it to all the messages he has in store—lo and behold, one of them produces a bit string that he can decrypt using his private key. Willie can now also tell that one of the messages he forwarded from Alice to Bob contained suspicious content, namely a random looking string with an error correction code attached, that was most likely an instance of the protocol described here. However, he cannot tell that the message was directed specifically to Bob, as he does not possess Bob's private key.

We have changed our model slightly, in that we now assume that Alice can send short pseudorandom messages with integrity. Otherwise, when Willie sees the message with the one-time key and realizes its significance, he would corrupt it or refuse to forward it. However, there might be circumstances in which he is unable to do this. For example, Alice might be at liberty while Bob is in jail; and Willie might be able to censor Alice's usenet postings via the prison's news server, but not permitted to censor comments that she makes from time to time on radio programs.

VII. Conclusion

We have explored the limits of steganographic theory and practice. We started off by outlining a number of techniques both ancient and modern, together with attacks on them (some new); we then discussed a number of possible approaches to a theory of the subject. We pointed out the difficulties that stand in the way of a theory of "perfect covertness" with the same power as Shannon's theory of perfect secrecy. But considerations of entropy give us some quantitative leverage and the "selection channel"—the bandwidth of the stego key—led us to suggest embedding information in parity checks rather than in the data directly. This approach gives improved efficiency, and also allows us to do public key steganography.

Finally, we have shown that public key steganography may sometimes be possible in the presence of an active warden.

Acknowledgment

Some of the ideas presented here were clarified by discussion with D. Wheeler, J. Daugman, R. Needham, G. Simmons, M. Kuhn, P. Rayner, D. Aucsmith, J. Kelsey, I. Jackson, M. Roe, M. Lomas, S. Lee, P. Wayner, M. Blaze, and S. Craver.

References

[1] E. Ambikairajah, A. G. Davis, and W. T. K. Wong, "Auditory masking and MPEG-1 audio compression," *IEE Electron. & Commun. Eng. J.*, vol. 9, no. 4, pp. 165–175, Aug. 1997.

[2] R. J. Anderson, "Liability and computer security: Nine principles," in *Computer Security—ESORICS'94*, vol. 875, pp. 231–245.

[3] ——, "Stretching the limits of steganography," in *Information Hiding, Springer Lecture Notes in Computer Science*, vol. 1174, pp. 39–48, 1996.

[4] "The eternity service," in *Proc. Pragocrypt 96*, pp. 242–252.

[5] R. J. Anderson and M. G. Kuhn, "Tamper resistance—A cautionary note," in *Proc. 2nd Usenix Workshop Electron. Commerce*, Nov. 1996, pp. 1–11.

[6] R. Anderson and C. Manifavas, "Chameleon—A new kind of stream cipher," to appear in *Proc. 4th Workshop on Fast Software Encryption*, 1997.

[7] R. J. Anderson, S. Vaudenay, B. Preneel, and K. Nyberg, *The Newton Channel*, this volume.

[8] D. Aucsmith, "Tamper resistant software: An implementation," in *Information Hiding, Springer Lecture Notes in Computer Science*, vol. 1174, pp. 317–333, 1996.

[9] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Syst. J.*, vol. 35, nos. 3–4, pp. 313–336, 1996.

[10] F. M. Boland, J. J. K. Ó Ruanaidh, and C. Dautzenberg, "Watermarking digital images for copyright protection," in *Proc. IEE Int. Conf. Image Processing and Its Applications*, Edinburgh, Scotland, 1995.

[11] L. Boney, A. H. Tewfik, and K. N. Hamdy, "Digital watermarks for audio signals," in *IEEE Int. Conf. Multimedia Computing and Systems*, Hiroshima, Japan, June 17–23, 1996, pp. 473–480.

[12] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "A secure, robust watermark for multimedia," in *Information Hiding, Springer Lecture Notes in Computer Science*, vol. 1174, pp. 183–206, 1996.

[13] R. Cox, presented at the Access All Areas Conference, London, U.K., May 7, 1997.

[14] S. Craver, "On public-key steganography in the presence of an active warden," IBM Res. Rep. RC 20931, July 23, 1997.

[15] E. Franz, A. Jerichow, S. Möller, A. Pfitzmann, and I. Stierand, "Computer based steganography," in *Information Hiding, Springer Lecture Notes in Computer Science*, vol. 1174, pp. 7–21, 1996.

[16] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, "Hiding routing information," in *Information Hiding, Springer Lecture Notes in Computer Science*, vol. 1174, pp. 137–150, 1996.

[17] D. Gruhl, A. Lu, and W. Bender, "Echo hiding," in *Information Hiding, Springer Lecture Notes in Computer Science*, vol. 1174, pp. 295–315, 1996.

[18] J. Gurnsey, *Copyright Theft*. Aslib Gower, 1995.

[19] R. Hart, "A voluntary international numbering system—The latest WIPO proposals," *Computer Law and Security Report*, vol. 11, no. 3, pp. 127–129, May–June, 1995.

[20] J. N. Holmes, *Speech Synthesis and Recognition—Aspects of Information Technology*. London, U.K.: Chapman & Hall, 1993.

[21] Talk on software birthmarks, counsel for IBM Corporation, BCS Technology of Software Protection Special Interest Group, London 1985.

[22] G. Jagpal, "Steganography in digital images," *Thesis*, Cambridge Univ. Comput. Laboratory, Cambridge, Univ. Cambridge, U.K., May 1995.

[23] D. Kahn, *The Codebreakers*. New York: Macmillan, 1967.

[24] A. Kerckhoffs, "La cryptographie militaire," *J. des Sciences Militaires*, ser. 9, no. IX, pp. 5–38, Jan. 1883, pp. 161–191, Feb. 1883.

[25] E. Koch and J. Zhao, "Toward robust and hidden image copyright labeling," in *Proc. 1995 IEEE Workshop on Nonlinear Signal and Image Processing*, Halkidiki, Greece, June 20–22, 1995.

[26] H. M. Kriz, "Phreaking recognized by directorate general of France telecom," *Chaos Digest 1.03*, Jan. 1993.

[27] C. Kurak and J. McHugh, "A cautionary note on image downgrading," in *IEEE Computer Security Applications Conf.*, 1992, pp. 153–159.

[28] S. Landau, S. Kent, C. Brooks, S. Charney, D. Denning, W. Diffie, A. Lauck, D. Miller, P. Neumann, and D. Sobel, "Codes, Keys and Conflicts: Issues in U.S. Crypto Policy," Rep. of a Special Panel of the ACM U.S. Public Policy Committee, June 1994.

[29] G. C. Langelaar, J. C. A. van der Lubbe, and J. Biemond, "Copy protection for multimedia data based on labeling techniques," presented at the *17th Symp. on Information Theory in the Benelux*, Enschede, The Netherlands, May 1996.

[30] N. F. Maxemchuk, "Electronic document distribution," *AT&T Tech. J.*, vol. 73, no. 5, pp. 73–80, Sept./Oct. 1994.

[31] B. C. J. Moore, *An Introduction to the Psychology of Hearing*. New York: Academic, 1989.

[32] I. S. Moskowitz and M. H. Kang, "Covert channels—Here to stay?" *Compass 94*, pp. 235–243.

[33] R. M. Needham, private communication, Dec. 1995.

[34] T. Parson, *Voice and Speech Processing*. New York: McGraw-Hill, 1986.

[35] B. Pfitzmann, "Information hiding terminology," in *Information Hiding, Springer Lecture Notes in Computer Science*, vol. 1174, pp. 347–350, 1996.

[36] ——, "Trials of traced traitors," in *Information Hiding, Springer Lecture Notes in Computer Science*, vol. 1174, pp. 49–64, 1996.

[37] I. Pitas, "A method for signature casting on digital images," in *Int. Conf. Image Processing*, vol. 3, Sept. 1996, pp. 215–218.

[38] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for web transactions," *DIMACS*, Tech. Rep. 97-15, Apr. 1997.

[39] D. L. Schilling, *Meteor Burst Communications: Theory and Practice*. New York: Wiley, 1993.

[40] B. Schneier, *Applied Cryptography—Protocols, Algorithms and Source Code in C*, 2nd ed. New York: Wiley, 1995.

[41] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423; 623–656, 1948.

[42] ——, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, pp. 656–715, 1949.

[43] G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Proc. CRYPTO'83*, 1984, pp. 51–67.

[44] ——, "How to insure that data acquired to verify treaty compliance are trustworthy," *Proc. IEEE*, vol. 76, p. 5, 1984.

[45] ——, "A survey of information authentication," in *Contemporary Cryptology—The Science of Information Integrity*. New York: IEEE Press, pp. 379–419, 1992.

[46] ——, "The history of subliminal channels," this issue, pp. 452–462.

[47] N. van Someren, "High quality de-interlacing of television images," Ph.D. Thesis, University of Cambridge, Cambridge, U.K., Sept. 1994.

[48] K. S. Jones, private communication, Aug. 1995.

[49] "Police to shut out snoopers," *Sunday Times*, p. 3.13, July 13, 1997.

[50] J. T. Trostle, "Modeling a fuzzy time system," in *Proc. IEEE Symp. in Security and Privacy 93*, pp. 82–89.

[51] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *IEEE Int. Conf. Image Processing*, 1994, vol. 2, pp. 86–90.

[52] P. Wayner, *Disappearing Cryptography—Being and Nothing on the Net*, AP Professional, 1996.

[53] K. Wong, "Fighting mobile phone fraud—Who is winning?," in *Datenschutz und Datensicherheit*, pp. 349–355, June 1995.

Ross J. Anderson, for a photograph and biography, see this issue, p. 450

Fabien A. P. Petitcolas graduated from the École Centrale, Lyon, France, and received the Diploma in computer science from the University of Cambridge, U.K.

He is currently a research student at the Computer Laboratory, University of Cambridge. His research topic is the robustness of information hiding systems.