Stretching the Limits of Steganography

Ross Anderson

Cambridge University Computer Laboratory
Pembroke Street, Cambridge CB2 3QG, UK
Email `rja14@cl.cam.ac.uk`

Abstract. We present a number of insights into information hiding. It was widely believed that public key steganography was impossible; we show how to do it. We then look at a number of possible approaches to the theoretical security of hidden communications. This turns out to hinge on the inefficiency of practical compression algorithms, and one of the most important parameters is whether the opponent is active or passive (i.e., whether the censor can add noise, or will merely allow or disallow a whole messages). However, there are covertexts whose compression characteristics are such that even an active opponent cannot always eliminate hidden channels completely.

1 Introduction

Steganography is about concealing the existence of messages, and it goes back to ancient times. Kahn tells of a classical Chinese practice of embedding a code ideogram at a prearranged place in a dispatch; of the warning the Greeks received of Xerxes' intentions from a message underneath the wax of a writing tablet; and a trick of dotting successive letters in a covertext with secret ink, due to Aeneas the Tactician [8].

The opponent may be passive, and merely observe the covertext, but he may also be active. In the US post office during the second world war, postal censors deleted lovers' X's, shifted watch hands, and replaced items such as loose stamps and blank paper. They also rephrased telegrams; in one case, a censor changed 'father is dead' to 'father is deceased', which elicited the reply 'is father dead or deceased?'

The study of this subject in the open scientific literature may be traced to Simmons, who in 1983 formulated it as the prisoners' problem [16]: Alice and Bob are in jail, and wish to hatch an escape plan. All their communications pass through the warden, Willy. If Willy sees any encrypted messages, he will frustrate their plan by putting 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 rely solely on a secret key.

There are many real life applications of steganography. Apparently, during the 1980's, British Prime Minister 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 of documents, 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 [10].

Simmons' real application was more exotic — the verification of nuclear arms control treaties. The US 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 out the forbidden information past the other country's monitoring facilities [17, 19].

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.

However, we still have no comprehensive theory of steganography, in the way that Shannon gave us a theory of encryption [15] and Simmons of authentication [18]. In this article, we will try to move a few small steps towards such a theory.

2 The State of the Art

A number of computer programs are available that will embed a ciphertext file in an image. The better systems assume that both sender and receiver share a key and use a conventional cryptographic keystream generator [13] to expand this into a long pseudo-random keystream. The keystream is then used to select pixels in which the bits of the ciphertext are embedded.

Of course, not every pixel may be suitable for encoding ciphertext: changes to pixels in large fields of monochrome colour, or that lie on sharply defined boundaries, might be visible. So some systems have an algorithm that determines whether a candidate pixel can be used by checking that the variance in luminosity of the eight surrounding pixels is neither very high (as on a boundary) nor very low (as in a monochrome field). A bit can be embedded in a pixel that passes this test by some rule such as setting its low order bit to the parity of the surrounding pixels (though in practice one might use something slightly more complicated to avoid leaving telltale statistics).

Of course, the more bits per pixel, the less correlated the low order bits will be with neighbouring bits and with higher order bits in the same pixel. Some quantitative measurements of the correlations between pixels on different bit planes in digital video may be found in [20]. In effect, the bits that Alice can use to embed covert data are redundant in that Willy will be unaware that they have been altered. It follows that they might be removed by an efficient compression scheme, if one exists for the image or other covertext in use.

So when the image is to be subjected to compression (whether before or after the insertion of covert material), things become more complicated, and we have to tailor the embedding method. For example, with `.gif` files one can swap colours for similar colours that are adjacent in the current palette [7], 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 [9] or in the frequency domain by altering components of the image's discrete cosine transform [3] [23]. Further papers on the topic may be found in this volume.

So the general model is that Alice embeds information by tweaking some bits of some transform of the covertext. The transform enables her to get at one or more bits which are redundant in the sense that tweaking them cannot be detected easily or at all. To a first approximation, we will expect that such transforms will be similar to those used for compression, and that there are many low-bandwidth stego channels arising from redundancy whose elimination, by compression or otherwise, is uneconomic for normal users of the cover system. We will not expect to find many high bandwidth channels, as these would normally correspond to redundancy that could economically be removed.

3 Public Key Steganography

So far, we have merely stated the general intuition of people who have thought about these topics. They generally assume that steganography, in the presence of a capable motivated opponent who is aware of the general methods that might be used, requires the pre-existence of a shared secret so that the two communicating parties can decide on which bits to tweak. So there has been a general assumption that public-key steganography is impossible.

However, this is not the case. We will now show how a hidden message can be sent to a recipient with whom the sender has no shared secret, but for whom an authentic public key is available.

Given a covertext in which any ciphertext at all can be embedded, then there will usually be a certain rate at which its bits can be tweaked without the warden noticing (we will discuss this more fully below). So suppose that Alice can modify at least one out of every hundred bits of the covertext. This means that Willy cannot distinguish the parity of each successive block of a hundred bits from random noise, and it follows that she can encode an arbitrary pseudorandom string in these parities.

This pseudorandom material will lie in plain sight; anyone will be able to read it. So Willy cannot simply check a covertext by seeing whether a pseudorandom string can be found in it. Indeed, a suitable parity check function will extract pseudorandom-looking data from any message in which covert information can be inserted at all.

Now suppose that Alice and Bob did not have the opportunity to agree 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 as described elsewhere in this volume.

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 with previously shared secrets). Public key stego scales less well than public key crypto, as every recipient has to try to decrypt every message. However, this appears to be an intrinsic property of anonymous communications.

4 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 encodes ciphertext at a very low rate. For example, the kth bit of ciphertext might become 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 intuitively obvious fact could be rigorously proved?

This leads us to ask what a proof of perfect covertness would look like. A working definition of of a secure stegosystem might be one for which Willy cannot differentiate between raw covertext and the stegotext containing embedded information, unless he has knowledge of the key. As in the case of cryptography, we might take Willy 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 message, so the system must generate enough plausible 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 dependent on the model of the source. There are a number of ways in which we can tackle this dependence, and we will present three of them. It is an open question whether any of them will yield useful results in any given application.

4.1 Selection channel

Our first idea is inspired by the correction channel that Shannon uses to prove his second coding theorem. This is the channel which someone who can see both the transmitted and received signals uses to tell the receiver which bits to tweak, and produces various noise and error correction bounds [14].

In a similar way, when Alice and Bob use a shared one-time pad to decide which covertext bit will contain 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 which 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) [8]. The book cipher is just a selection channel. The model of computation may be different, in that with a book cipher we start off with the book and then generate the ciphertext, whereas in a stegosystem, we start off with the text to be embedded and then create the stegotext; but they are clearly related.

A repetitive book will have a lower capacity, as we will be able to use a smaller percentage of its words before inference 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.

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 information theoretic approach suggests a better way. We will use our 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 the covertext is '1' with probability 0.6, and we encode in bit pairs, 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. As the improvement is geometric, we can with little difficulty get the discrepancy as low as we like and thus — provided that our selection channel is broad enough — we can hide information quite efficiently.

Indeed, there is a tradeoff: the more bits in the selection channel, 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.

But clearly there is an ultimate limit to the amount of information that can be hidden.

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. Then the covert capacity will be at most $H(E) - H(M)$. Can we tighten this upper bound or — better still — get any kind of lower bound?

4.2 Equivalence classes

We can consider equivalence classes of messages. Suppose Alice uses a keyed cryptographic hash function to derive one bit from each sentence of a letter. As she prepares her stegotext letter to Bob, she has a routine which checks this bit and beeps if it is wrong. This will go off about every other sentence, which she can then rewrite.

Of course, if she uses standard changes such as [is able $\leftrightarrow$ can], then clearly she must not alter their statistics to the point that Willy 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 [21] — that is, embed a ciphertext using the technique described above — and the problem is commended to the research community as a weak form of the Turing test.

The converse is that writing a program to scan for human inserted steganography might be rather hard. This is an area in which competitive research might prove interesting!

Equivalence classes can also arise naturally. For example, when making a map from a larger scale map, many arbitrary decisions have to be taken about which features to incorporate, especially with features such as coastlines that are to some extent fractal [12]. 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 [6].

Equivalence classes of messages are tied up with compression. If covertext C_1 means the same as 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 so the compression is lossy.

Information theorists assume that any signal can in theory be completely compressed. But if this could ever be done in practice, then the steganography problem would become trivial: Alice can just 'uncompress' her ciphertext getting a comprehensible message, and Willy would have to pass the result. So Willy can only do his job at all because perfect compression is unfeasible. So regardless of whether we talk in terms of compression or of equivalence classes, Willy's performance of his job depends on his having a model of the source, and the danger to Alice and Bob is that his model might be better than theirs.

4.3 Conditional entropy

If Alice does not trust her ability to spot which bits of the covertext (or some transform of it) will appear noise-like to Willy, she might try to use conditional entropy. If the ciphertext is random, its mutual information with the covertext will be zero, and entropy will be additive — the entropy of the loaded covertext will be exactly equal to the entropy of the pure covertext plus that of the ciphertext.

Thus the maximum amount of ciphertext that she can hide from Willy depends on the uncertainty of the covertext entropy. Confidence levels can be computed in the obvious way: if the ciphertext rate is R_1, then the warden must not be able to detect with probability P a change of R_1 in the covertext rate R_2.

Thanks to the Central Limit Theorem, the more covertext we give the warden, the better he will be able to estimate its statistics, 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 [11]. However, as a matter of empirical fact, there do exist channels in which ciphertext can be inserted at a positive rate [4], so measuring entropy may be useful in a number of applications.

However, it still does not give us a way to prove the unconditional covertness of a system. The reason for this is that once Alice assumes that Willy is smarter than she is, she has no way of estimating the variance in his estimates of the entropy of her covertext. A purist might conclude that the only circumstance in which she can be certain that Willy cannot detect her messages is when she uses a subliminal channel in the sense of Simmons; that is, a channel in which she chooses some random bits (as in an ElGamal digital signature) and these bits can be recovered by the message recipient [1].

5 Active and Passive Wardens

The applications discussed above include both passive wardens, who monitor traffic and signal to some process outside the system if unauthorised message traffic is detected, and active wardens who try to remove all possible covert messages from covertexts that pass through their hands. A good example of the latter was the world war two postal censor described in the introduction, and a highly topical example is given by software piracy.

Software birthmarks, as mentioned above, have been used to prove the authorship of code so that pirates could be prosecuted. They were serviceable with hand assembled system software, but might be harder to find now that most code is produced by a compiler. A possible remedy is to embed copyright information by mangling the object code in some way. The automatic, random replacement of code fragments with equivalent ones is used by Intel to customise security code [2]. This may be adequate in that application, where the goal is to prevent a single patch defeating all instances of a protective mechanism; but copyright marking is harder. One could imagine a contest between software authors and pirates to see who can mangle code most thoroughly without affecting 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.

In fact, the World Intellectual Property Organisation has proposed a system of numbering for all digital works, including books, sound and video recordings, and computer programs; it claims that the boundaries between these are breaking down. Software publishers are sceptical; they claim to have had no difficulty yet in establishing ownership [5]. But whatever the legal value of copyright marking, the software pirate is a good example of an active warden.

In such a case, the simple public key scheme described in section two above will not work. Even in the shared-key model, there are cases where an active warden can completely block the stego channel. For example, if (a) his model of the communication at least as good as the prisoners' (b) the covertext information separates cleanly from the covert information, then he can replace the latter with noise. This is the case of a software pirate who has a better code mangler than the software author.

6 Limits on Active Wardens

However, there are many other cases where the stego channel is highly bound up with the covertext. For example, Jagpal [7] measured the noise that can be added to a `.gif` file before the image quality is degraded, while Möller and others have done the same for digitised speech [4].

The point here is that if Alice can add an extra X% of noise without affecting the picture, then so can Willy; but she can stop him finding out which X% carries the covert message by using a keystream to select which bits of covertext to tweak. In this case, all Willy will be able to do is to cut the bandwidth of the channel — a scenario that Trostle and others have explored in the context of covert channels in operating systems [22].

This bandwidth limitation will also be effective against systems that embed each ciphertext bit as a parity check of a number of covertext 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 covertext.

It is an open question whether public key steganography can be made to work against an active warden who can add only a limited amount of noise. It may also be of interest to consider whether one can implement other cryptographic primitives, such as the wiretap channel and bit commitment [13]. If it turns out that the kind of public key steganography that we have described here cannot be made to work, then key exchange well might be possible by combining techniques like these.

7 Conclusions

We have stretched the limits of steganography somewhat. Firstly, we have shown how to do public key steganography. Secondly, we have discussed a number of possible approaches to a theory of the subject, which suggest various practical techniques for improving the covertness of existing steganographic schemes. Thirdly, we have highlighted one of the most important topics, namely whether the warden is active or passive, and shown how this interacts with both the public key and theoretical approaches to the subject.

Acknowledgements: Some of the ideas presented here were clarified by discussion with David Wheeler, John Daugman, Roger Needham, Gus Simmons, Markus Kuhn, John Kelsey, Ian Jackson, Mike Roe, Mark Lomas, Stewart Lee. Peter Wayner and Matt Blaze. I am also grateful to the Isaac Newton Institute for hospitality while this paper was being written.

References

1. "The Newton Channel", RJ Anderson, S Vaudenay, B Preneel, K Nyberg, *this volume*
2. "Tamper Resistant Software: An Implementation", D Aucsmith, *this volume*
3. "Watermarking Digital Images for Copyright Protection", FM Boland, JJK Ó Ruanaidh, C Dautzenberg, *Proceedings, IEE International Conference on Image Processing and its Applications, Edinburgh 1995*
4. "Computer Based Steganography", E Franz, A Jerichow, S Moeller, A Pfitzmann, I Stierand, *this volume*
5. "A voluntary international numbering system — the latest WIPO proposals", R Hart, *Computer Law and Security Report* v 11 no 3 (May-June 95) pp 127–129
6. Talk on software birthmarks, counsel for IBM Corporation, BCS Technology of Software Protection Special Interest Group, London 1985
7. *'Steganography in Digital Images'*, G Jagpal, Thesis, Cambridge University Computer Laboratory, May 1995
8. *'The Codebreakers'*, D Kahn, Macmillan 1967
9. "Towards Robust and Hidden Image Copyright Labeling", E Koch, J Zhao, *Proceedings of 1995 IEEE Workshop on Nonlinear Signal and Image Processing* (Neos Marmaras, Halkidiki, Greece, June 20–22, 1995)
10. "Electronic Document Distribution", NF Maxemchuk, *AT & T Technical Journal* v 73 no 5 (Sep/Oct 94) pp 73–80
11. "Covert Channels — Here to Stay?", IS Moskowitz, MH Kang, *Compass 94* pp 235–243
12. RM Needham, *private conversation*, December 1995
13. *'Applied Cryptography — Protocols, Algorithms and Source Code in C'* B Schneier (second edition), Wiley 1995

14. "A Mathematical Theory of Communication", CE Shannon, in *Bell Systems Technical Journal* v 27 (1948) pp 379–423, 623–656
15. "Communication theory of secrecy systems", CE Shannon, in *Bell Systems Technical Journal* v 28 (1949) pp 656–715
16. "The Prisoners' Problem and the Subliminal Channel", GJ Simmons, in *Proceedings of CRYPTO '83*, Plenum Press (1984) pp 51–67
17. "How to Insure that Data Acquired to Verify Treaty Compliance are Trustworthy", GJ Simmons, *Proceedings of the IEEE* v 76 (1984) p 5
18. "A survey of information authentication", GJ Simmons, in *Contemporary Cryptology — the Science of information Integrity*, IEEE Press 1992, pp 379–419
19. "The History of Subliminal Channels", GJ Simmons, *this volume*
20. *'High Quality De-interlacing of Television Images'*, N van Someren, PhD Thesis, University of Cambridge, September 1994
21. K Spärck Jones, *private communication*, August 1995
22. "Modelling a Fuzzy Time System", JT Trostle, *Proc. IEEE Symposium in Security and Privacy 93* pp 82 - 89
23. "Embedding Robust Labels Into Images For Copyright Protection", J Zhao, E Koch, *Proc. Int. Congr. on IPR for Specialized Information, Knowledge and New Technologies* (Vienna, Austria, August 21-25, 1995)