PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 44, NO. 3–4, PP. 249–258 (2000)

STEGANOGRAPHIC METHODS

József LENTI
Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
H–1521 Budapest, Hungary

Received: June 5, 2000

Abstract

In this paper we analyze and test several steganographic techniques on still images. We show that embedding a large amount of data into the picture it can modify its visible properties. We compare the RSA and the elliptic curve (ECC) based digital signatures, and we analyze their advantages and disadvantages in steganography. In steganography it is important that the embedded data size should be minimized. Using ECC minimization of the embedded information is possible, because the minimal block size is smaller than in the case of RSA.

Keywords: security, imaging, steganography.

1. Introduction

Digital communication has become an essential part of infrastructure nowadays, a lot of applications are Internet-based and in some cases it is desired that the communication be made secret. Two techniques are available to achieve this goal: one is cryptography, where the sender uses an encryption key to scramble the message, this scrambled message is transmitted through the insecure public channel, and the reconstruction of the original, unencrypted message is possible only if the receiver has the appropriate decryption key. The second method is steganography, where the secret message is embedded in another message. Using this technology even the fact that a secret is being transmitted has to be secret [2].

There are two main directions in information hiding: protecting only against the detection of a secret message by a passive adversary, and hiding data so that even an active adversary cannot remove it. The classic situation, known as Simmons' "Prisoners' Problem", is the following: Alice and Bob are in jail and try to discuss an escape plan, but all their communication can be observed by the warden. If their plan or the fact that they are discussing an escape plan were detected they would be transferred to a more secure prison. So they can only succeed if Alice can send messages to Bob so that the warden can't even detect the presence of a secret [8].

There are a lot of real applications of steganography. For example during the 80s some confidential cabinet documents were passed to the English press so Margaret Thatcher had the word processors modified to encode the identity of the user into the word spacing of the documents so the identity of an information source could be found out [2].

In this article we show some possible steganographic methods on still images in section 2 – how to hide data in image files, and some experiments using steganographic methods on test pictures, and we make a proposal to improve the embedding process using a smaller size digital signature, but providing the same level of security.

2. Requirements

There are different requirements depending on the purpose of steganography:

- *Capacity*: it is an important factor in captioning applications, when a lot of information should be embedded into a cover image, what is usually related to the current picture. For example when transmitting medical images, the personal data, and the diagnosis could be embedded into the same picture.
- *Imperceptibility*: it is important when a secret communication occurs between two parties and the fact of a secret communication is kept to be secret.
- *Robustness*: watermarking, fingerprinting and all copyright protecting applications demand robust steganographic method, i.e. where the embedded information cannot be removed without serious degradation of the image [12].

3. Survey of Methods, Experiments

3.1. The Embedding Process

Steganography embeds a secret message in a cover message, this process is usually parameterized by a stego-key, and the detection or reading of an embedded information is possible only having this key. *Fig. 1* shows this process.

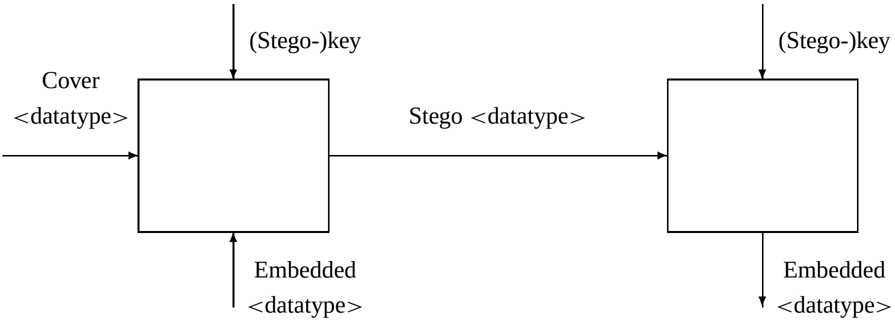


Fig. 1.

3.2. *Fingerprinting and Watermarking*

Nowadays steganography is more and more important in publishing and broadcasting industries, where the embedding of copyright marks or serial numbers is needed in digital films, photos and other multimedia products. Some steganographic applications are able to scan the Internet, and to detect a copy of a specific image, or the modified image is published – so an illegal usage of a copyrighted image can be detected. In the case of audio materials, the automatic monitoring of radio advertisements is also possible, the advertiser can automatically count how many times a specific advertisement was transmitted by a given radio station. Another possible application in the case of still images is to embed captions and other information into the picture so that one does not have to store distinctly the images, and connected information.

When the purpose is the protection of intellectual property, we can make a distinction between *fingerprinting* and *watermarking*. In the case of watermarking a copyright information is embedded in a digital media, and this media is transmitted to users. Fingerprinting embeds separate mark in the copies of digital media, this embedded information serves as a serial number, it can be detected who supplied this media to third parties.

3.3. *Least Significant Bit Insertion*

Usually 24-bit or 8-bit files are used to store digital images. The former one provides more space for information hiding, however, it can be quite large. The colored representations of the pixels are derived from three primary colors: red, green and blue. 24-bit images use 3 bytes for each pixel, where each primary color is represented by 1 byte. Using 24-bit images each pixel can represent 16,777,216 color values. We can use the lower two bits of these color channels to hide data, then the maximum color change in a pixel could be of 64-color values, but this causes so little change that is undetectable for the human vision system. This simple method is known as *Least Significant Bit insertion* [4], [15]. Using this method it is possible to embed a significant amount of information with no visible degradation of the cover image. *Fig. 2* shows the process.

Several versions of LSB insertion exist. It is possible to use a random number generator initialized with a stego-key and its output is combined with the input data, and this is embedded to a cover image. For example in the presence of an active warden it is not enough to embed a message in a known place (or in a known sequence of bits) because the warden is able to modify these bits, even if he can't decide whether there is a secret message or not, or he can't read it because it is encrypted. The usage of a stego-key is important, because the security of a protection system should not be based on the secrecy of the algorithm itself, instead of the choice of a secret key [11]. *Fig. 3* shows this process.

The LSB inserting usually operates on bitmap images. 'Steganos for Win-

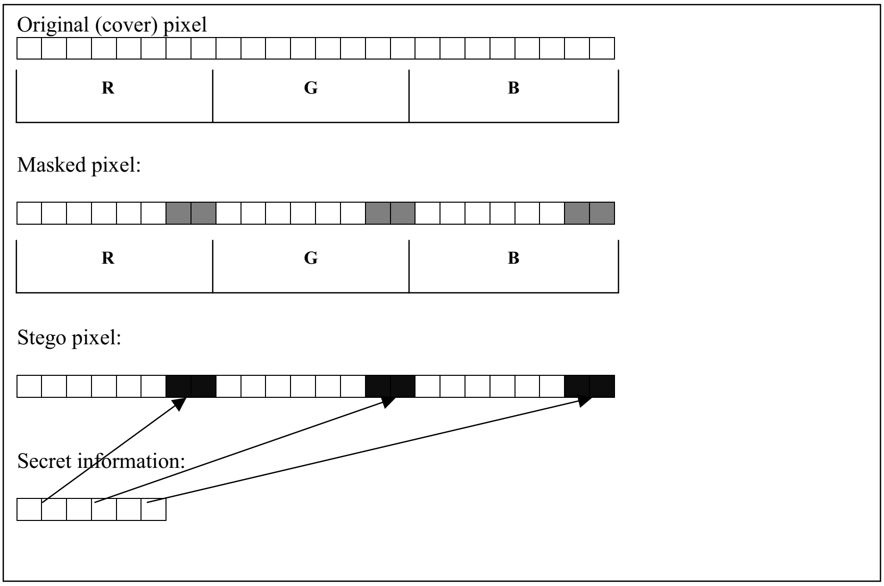


Fig. 2.

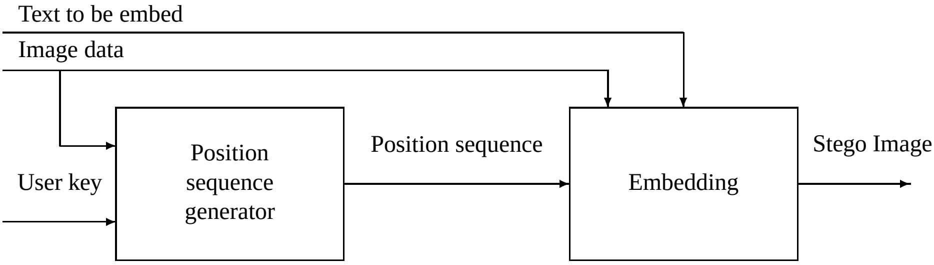
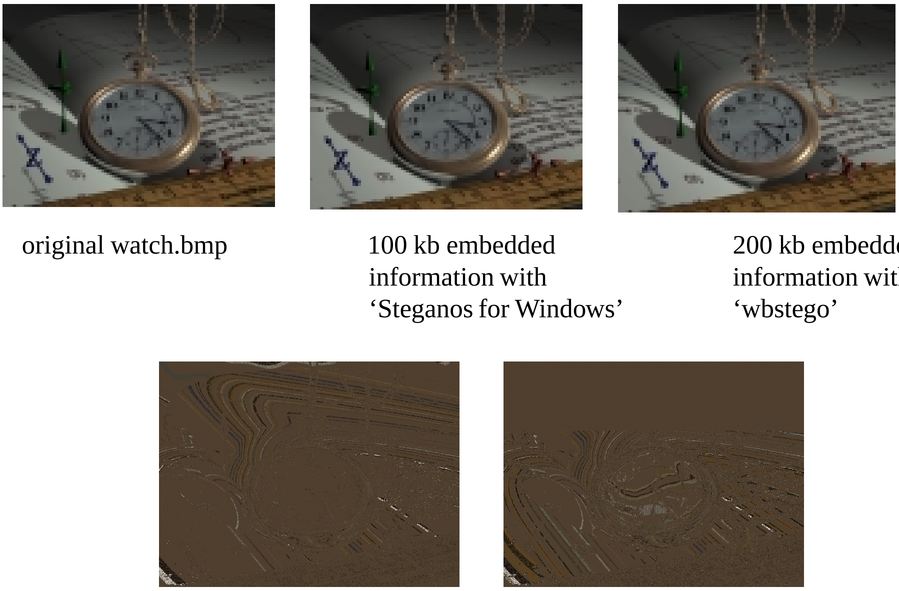


Fig. 3.

dows' and 'Wbstego' are LSB inserting software products which are able to embed data (in clear or encrypted format) in a bitmap image. The embedded data cannot be considered as a watermark, because even if a small change occurs in a picture (cropping, lossy compression, color degradation) the embedded information will be lost – although the change which is occurred during the embedding process is invisible.

The original bitmap picture which was used during the test was a picture 1024 × 768 pixels in size, with 16M colors (it is a standard test picture in image processing). We made a test using bitmap images. The following pictures will show the results using different software with different embedded data size:



original watch.bmp

100 kb embedded information with ‘Steganos for Windows’

200 kb embedded information with ‘wbstego’

the difference between the original and the modified (100 kb Steganos for Win.)

the difference between the original and the modified (200 kb wbstego)

Fig. 4.

When these pictures were modified all the embedded information was lost. These softwares do not use any redundancies during embedding, the embedding process does not apply any error correcting codes. In this case the error correction and the redundancies are useful only if the image is modified in bmp format. If a lossy compression technique is applied, usually all the lsb bits are lost, therefore all embedded information is also destroyed.

3.4. Public Key Steganography

As another possible way the algorithm requires the pre-existence of a shared secret key to designate pixels which should be tweaked. In this case both the sender and the receiver must have this secret. Suppose that the communicating parties do not have the opportunity to agree a secret key, but one of them (e.g. Bob) has a private/public key pair, and his partner knows the public key. In the case of a passive warden Alice knowing Bob’s public key encrypts her message with this key, embeds it in a known channel (known position in the cover media), and sends it to Bob. Bob cannot be sure whether the channel contains a hidden message, but he can try to decrypt the random-looking string-sequence with his private key, and check whether it is a message or not [5].

Another approach is the *cover image escrow scheme* (or source extraction), where the extractor is required with the original cover image, and the cover image is subtracted from the stego image before the extraction of the embedded information. In this scheme, the user cannot read the embedded data, it is only possible to have the original unmodified picture, but these types of algorithms are characterized as robust against signal distortions.

3.5. Transform Domain Based Steganography

The destination extraction algorithms can be divided into two groups: spatial/time domain and transform domain techniques. In the former case information is embedded in the spatial domain in the case of images, and in time domain in the case of audio materials. The transform domain methods operate in the Discrete Cosine Transform, Fourier or wavelet transform domains of the host signal [2], [11], [15].

The *Patchwork algorithm* (developed at the MIT) selects random pairs of pixels, and increases the brightness of the brighter pixel and decreases the brightness of the other. This algorithm shows a high resistance to most nongeometric image modifications. If it is important to provide a protection against filtering attacks, then the information hiding capacity is limited [4].

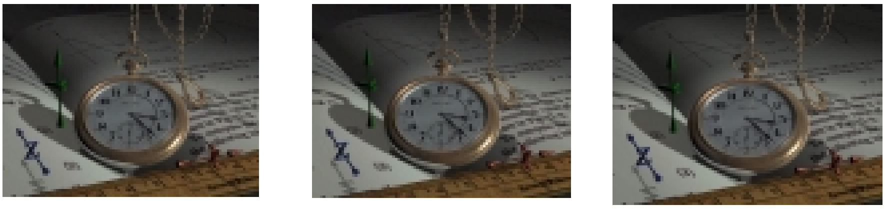
High color quality images are compressed usually using a lossy compression method as, for example, in the case of Jpeg images. In Jpeg algorithm the pixels are first transformed into a luminance-chrominance space. The chrominance is then downsampled – it is possible because the HVS (Human Vision System) is less sensitive to chrominance changes than to luminance changes – so the volume of the data is reduced. Discrete Cosine Transform is then applied on the groups of 8 × 8 pixels. The next step causes the most loss in the case of Jpeg, where the coefficients are scalarly quantized (it is possible because if we reduce the coefficients of higher frequencies to zero, the changes to the original image will cause only small changes that a human viewer could not detect under normal circumstances). The final steps are lossless, when these reduced coefficients are also compressed and a header is added to the Jpeg image. (See a detailed description in [5]). Steganographic applications usually operate after the quantization step, for example Jpeg-Jsteg, and SysCoP. SysCoP uses a position sequence generator. The inputs of the generator are the image data and user key, the output is a position sequence for selecting blocks where the code is embedded [14], [2].

The block consists in this case of 8 × 8 pixels, it can be contiguous – the block is a square in the image – or distributed, where the pixels are randomly selected. A label bit is embedded through setting specific relationship among three quantized elements of a block, and the algorithm contains a checking mechanism to test whether the actual block is capable or not to store this information, how big modification is needed to store one bit information among these pixels.

A popular method in a frequency domain is to modify the relative size of two or more DCT coefficients in an image block, embedding one bit information in each block. The two coefficients should correspond to cosine functions with middle frequencies which means that the information is stored in a significant part of the signal. The algorithm should be robust against Jpeg compression, so the DCT coefficients with equal quantization values should be chosen, according to the quantization table of Jpeg.

In the frequency domain the embedding process can usually hide less information into pictures, there is not such an exact limit in the size of the embedded object as in the case of LSB insertion, where the number of pixels, and the color depth determine the maximum size of the embedded data (and it was sure, that the changes occurred during embedding will be invisible).

In the case of a transform domain operation the embedding process can cause visible changes if the embedded data size is too big, and the limit where a given embedded data size does not change the visual properties of the image is image-dependent. The following figures show the result of the embedding process in transform domain.



30 kb of embedded data with ‘jhps’

50 kb of embedded data with ‘jhps’

60 kb of embedded data with ‘jhps’

Fig. 5.

In the case of a watch test picture 50 kb embedded data (and above) modifies the visible properties of the image, so when the stego-image is compared with the original one it is possible to recognize a modification.

4. Attacks

LSB insertion is an easy way and provides a high capacity to embed data into images, but it does not provide protection against small changes resulting from lossy compression or image transformations.

In the case of watermarking, a steganographic scheme should be able to resist some basic manipulations – which can be done using standard image manipulating tools – resampling, resizing, rotation, lossy compression.

Of course, there are available software tools against steganographic schemes, maybe the most complex one is StirMark, which was developed to test the steganographic algorithms on still images. StirMark can simulate a complex resampling process, when an image is printed and scanned with a high quality scanner and during this process some geometric distortions occur [5].

A much easier but still useful attack is the 'Mosaic attack'. The purpose of this attack is to prohibit the success of copyright detection software, which can be implemented as a WWW scanner, downloading images from web-sites and checking whether a watermark is embedded. This method chops the images into small pieces and a picture on a web page is edited from this small image-pieces, so an ordinary web browser would show them as a single image. Using this attack no degradation occurs on the original picture, there is no need for a huge computing power but it is still useful because all the schemes require a minimal picture size to be able to embed information into an image, so chopping the picture to smaller pieces than the minimal size required by the current algorithm prohibits the detection software the detection of the copyright information [6].

5. Improvement of the Embedding Process

When a watermark is embedded into a picture, it is a unique bit sequence which can prove the ownership of the image. The embedded data must have the property, that after extraction it is difficult to modify. Even if the attacker is able to extract a part of the embedded information, the modification of that data should invalidate the whole copyright information. For example, if the embedded copyright information would be the simple text: 'This watch picture belongs to Alice', the modification of the word 'Alice' should invalidate the watermark.

Another important property of the embedded watermark is the size. If the size is small – given an embedding capacity provided by the watermarking method – error correcting codes and redundancies can be applied so the security can be improved.

The current watermarking solutions are using a digital signature with a Secure Hash Algorithm, encrypted with 1024 bit RSA, so the signature size is also 1024 bits. If not a digital signature is used, the most commonly applied public-key encryption methods are the RSA, and the Discrete Logarithm-based El Gamal algorithm. The key size in the case of El Gamal is also 1024 which provides the same security level as RSA with 1024 bit key. The encrypted message size in the case of short – 100 bits – messages are 1024 bits for RSA, and 2048 for El Gamal.

The public key cryptosystem based on elliptic curves (ECC) can provide a much favourable output. Since it is considered the most secure per bit public key cryptosystem, it can provide the same security as RSA with a smaller key size.

The following picture shows the comparison of the required key sizes between RSA and the Elliptic Curve Cryptosystem (ECC) which is needed to provide the same level of security.

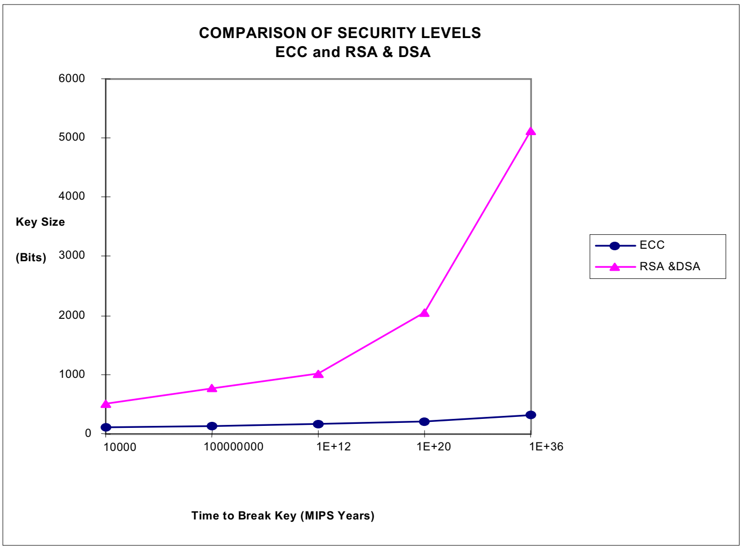


Fig. 6.

ECC over a 160 bit field GF(2^{160}) is equivalent with a 1024-bit modulus RSA. The signature size in the case of Elliptic Curve Digital Signature Algorithm is 320 bit long, and the encrypted message size in the case of 100 bit message is 321 bit over a 160 bit field.

It means that ECC can be more efficient in watermarking applications, because we need to embed less data so the probability that any visible changes occur during the embedding process is reduced. If we can embed higher amount of data than the minimal size determined by the encryption algorithm we can apply error correcting codes, or redundancies during embedding which provides robustness against possible image distortions.

6. Conclusion

In this paper several techniques are discussed how to embed information in still images. We discussed what are the possible requirements in data hiding, and what kinds of attacks are possible against steganographic methods.

We tested some steganographic software, we studied and analyzed the visible changes caused by the embedding process, and their resistance against distortions.

We proposed the usage of Elliptic Curves instead of RSA, because it is more efficient in case of steganographic applications, because of the small size of digital signatures and encrypted small messages using this technique.

References

[1] PETITCOLAS, F. A. P. – ANDERSON, R. J. – KUHN, M.G., Attacks on Copyright Marking Systems, In David Aucsmith, Ed., *Second Workshop on Information Hiding*, in Vol. 1525 of *Lecture Notes in Computer Science*, Portland, Oregon, U.S.A., 14–17 April, 1998, pp. 218–238. ISBN 3-540-65386-4.

[2] ANDERSON, R. J. – PETITCOLAS, F. A. P., On The Limits of Steganography, *IEEE Journal of Selected Areas in Communications*, **16** (4) pp. 474-481, May 1998. Special Issue on Copyright & Privacy Protection. ISSN 0733-8716.

[3] BRASSIL, J. – LOW, S. – MAXEMCHUK, N. F. – O'GORMAN, L., Hiding Information in Document Images, AT&T Bell Laboratories, Murray Hill, NJ.

[4] BENDER, W. – GRUHL, D. – MORIMOTO, N., Techniques for Data Hiding, Massachusetts Institute of Technology, Media Laboratory Cambridge, Massachusetts 02139 USA, From the *Proceedings of the SPIE*, 2420:40, San Jose CA, February, 1995.

[5] On Public-key Steganography in the Presence of an Active Warden, Scott Craver Intel Corporation Microcomputer Research Labs 2200 Mission College Blvd., Santa Clara, CA 95052–8119 Department of Mathematical Sciences Northern Illinois University DeKalb, IL 60115.

[6] JOHNSON, N. F. – JAJODIA, S., Exploring Steganography: Seeing the Unseen, George Mason University.

[7] KAWAGUCHI, E. – EASON, R. O., Principle and Applications of BPCS-Steganography *Proc. of 1998 SPIE International Symposium on Voice, Video, and Data Communications*, Conference #3528.

[8] CACHIN, C., An Information-Theoretic Model for Steganography, MIT Laboratory for Computer Science 545 Technology Square Cambridge, MA 02139, USA cachin@acm.org May 13, 1998 In *Proceedings of 2nd Workshop on Information Hiding* (D. Aucsmith, ed.), *Lecture Notes in Computer Science*, Springer, 1998.

[9] BENDER, W. – GRUHL, D. – MORIMOTO, N. – LU, A., Techniques for Data Hiding, *IBM Systems Journal*, **35**, Nos. 3–4, 1996.

[10] HWANG, R. W., Robust Algorithm for Information Hiding in Digital Pictures, Master of Engineering at The Massachusetts Institute of Technology May 21, 1999.

[11] ZHAO, J. – KOCH, E., Embedding Robust Labels Into Images For Copyright Protection, Fraunhofer Institute for Computer Graphics Wilhelminenstr. 7, 64283 Darmstadt, Germany Email: {zhao, ekoch}@igd.fhg.de *Proc. of the Int. Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*, Vienna, August 1995.

[12] KUNDUR, D. – HATZINAKOS, D., Mismatching Perceptual Models for Effective Watermarking in the Presence of Compression, Department of Electrical and Computer Engineering University of Toronto 10 King's College Road Toronto, Ontario Canada M5S 3G4.

[13] MITCHELL, L. – SWANSON, D. – ZHU, B. – TEWFIK, A. H., Robust Data Hiding for Images, Department of Electrical Engineering, University of Minnesota, Minneapolis, MN 55455 e mail: mswanson, binzhu, tewfik@ee.umn.edu In *IEEE Digital Signal Processing Workshop* (DSP 96), Loen, Norway, pp. 37–40, September, 1996.

[14] SMITH, J. R. – COMISKEY, B. O., fjrs, Modulation and Information Hiding in Images, elwoodg@media.mit.edu Physics and Media GroupMIT Media Lab 20 Ames Street Cambridge, MA 02139 USA *Proceedings of the First Information Hiding Workshop*, Isaac Newton Institute, Cambridge, U.K., May 1996. Springer-Verlag *Lecture Notes in Computer Science* Volume 1174.

[15] CURRIE, D. L., Surmounting the Effects of Lossy Compression on Steganography, III Fleet Information Warfare Center 2555 Amphibious Drive NAB Little Creek Norfolk, VA 23521 3225 currie@msn.comi Cynthia E. Irvine Computer Science Department Code CS/Ic Naval Postgraduate School Monterey, CA 93943 5118 irvine@cs.nps.navy.mil *Proceedings of the 19th National Information System Security Conference*, Baltimore, Md, October 1996, pp. 194–201.