Ira S. Moskowitz (Ed.)

LNCS 2137

Information Hiding

4th International Workshop, IH 2001
Pittsburgh, PA, USA, April 2001
Proceedings

Springer

Ira S. Moskowitz (Ed.)

Information Hiding

4th International Workshop, IH 2001
Pittsburgh, PA, USA, April 25-27, 2001
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editor

Ira S. Moskowitz
Naval Research Laboratory
Washington, DC 20375, USA

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Information hiding : 4th international workshop ; proceedings / IH 2001, Pittsburgh, PA, USA, April 25 - 27, 2001. Ira S. Moskowitz (ed.). – Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2137)
ISBN 3-540-42733-3

CR Subject Classification (1998):E.3, K.6.5, K.4.1, K.5.1, D.4.6, E.4, C.2, H.4.3

ISSN 0302-9743
ISBN 3-540-42733-3 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by DA-TeX Gerd Blumenstein
Printed on acid-free paper SPIN 10840135 06/3142 5 4 3 2 1 0

Preface

It is my pleasure and privilege to introduce the papers presented at the 4th International Information Hiding Workshop – IH 2001. We held the first meeting, which was chaired by Ross Anderson, at the Newton Institute, Cambridge, UK almost five years ago. At that meeting, as Ross stated in his introduction to the first proceedings, we initiated public discussion and critical analysis of five different approaches to information hiding problems: watermarking; anonymous communications; covert channels; steganography; and unobtrusive communications, such as spread-spectrum and meteor scatter radio. Our efforts to bring together, in one meeting, these diverse strands of the information hiding community proved successful, as have our subsequent meetings in Portland, Oregon, USA, under the chairmanship of David Aucsmith, and in Dresden, Germany, which was chaired by Andreas Pfitzmann.

Since our first meeting, the necessity that governments and businesses confront issues related to information hiding has not decreased. Rather, due in large part to the growth of the Internet, such concerns have become ever more urgent. Recently, for example, the news media has exposed the use of embedded information transfers by "undesirable" groups and hidden file structures. On a commercial level, the recent litigation over Napster is unlikely to foil the threats faced by owners of digitally-communicable data to their intellectual property rights. However, as our community recognizes, legitimate privacy concerns must also be respected in order for information hiding techniques to be recognized as both lawful and ethical. These issues make the research presented at IH 2001 even more pressing and timely.

As in previous years, researchers have approached issues related to the hiding of information from many different angles. For this workshop, we have made an effort to select papers which represent the gamut of interest to information hiders: watermarking and fingerprinting of digital audio, still image, and video; anonymous communications; steganography and subliminal channels; covert channels; database inference channels, etc. This year, several papers analyze problems related to chemistry and to natural language. On a more philosophical level, the papers also represent a mix of conjecture, theory, experimentation, and lessons learned.

We had many quality submissions this year. Unfortunately, due to the pressures of maintaining a balanced program and of providing each speaker with an adequate amount of time for presentation and discussion, we could accept only a small percentage of the submissions. In addition to the presented papers, we also had two discussion sessions. The difficult job of developing the program fell to the program committee which consisted of Ross Anderson (Cambridge University, UK), David Aucsmith (Intel Corp, USA), Jean-Paul Linnartz (Philips Research, The Netherlands), Steven Low (California Institute of Technology, USA), John McHugh(SEI/CERT, USA), Fabien Petitcolas (Microsoft Research, UK),

Andreas Pfitzmann (Dresden University of Technology, Germany), Jean-Jacques Quisquater (Université Catholique de Louvain, Belgium), Mike Reiter (Bell Labs, Lucent Technologies, USA) and Michael Waidner (IBM Zurich Research Lab, Switzerland), as well as myself. In addition, we are grateful for the assistance we received from Tuomas Aura, Oliver Berthold, LiWu Chang, Sebastian Clauß, Richard Clayton, George Danezis, Jean-François Delaigle, Cédric Fournet, Elke Franz, Teddy Furon, Ruth Heilizer, Markus Jakobsson, Anne-Marie Kermarrec, Darko Kirovski, Herbert Klimant, Stefan Köpsell, Garth Longdon, Henrique Malvar, Kai Rannenberg, and Jianxin Yan.

This year we split the chairpersonship into the positions of "general" chair and "program" chair. John McHugh was the general chair for IH 2001. Both he and his staff did a fantastic job with the local arrangements, putting together the preproceedings, and the registration process. In keeping with the nautical theme of the River Cam, the Columbia River, and the River Elbe, he arranged a wonderful dinner cruise for the workshop dinner. I thank John for the great job he has done!

If one looks through the past proceedings, in conjunction with IH 2001, it is exciting to see how the field of information hiding is growing and maturing. We are all looking forward to the new research that will be presented at the next workshop.

Finally, I would like to thank my colleagues on the program committee, the people who assisted the program committee, the workshop participants, and especially every author who submitted a paper to IH 2001. You all help make the workshop stronger and more interesting!

April 2001

Ira S. Moskowitz

Table of Contents

Trustworthy Paper Documents 1
Marshall Bern, Jeff Breidenbach, and David Goldberg

An Implementation of Key-Based Digital Signal Steganography 13
Toby Sharp

Distortion-Free Data Embedding for Images 27
Miroslav Goljan, Jessica J. Fridrich, and Rui Du

Information Hiding through Noisy Channels 42
Valeri Korjik and Guillermo Morales-Luna

A Perceptual Audio Hashing Algorithm:
A Tool for Robust Audio Identification and Information Hiding 51
M. Kıvanç Mıhçak and Ramarathnam Venkatesan

Computational Forensic Techniques for Intellectual Property Protection 66
Jennifer L. Wong, Darko Kirovski, and Miodrag Potkonjak

Intellectual Property Metering 81
Farinaz Koushanfar, Gang Qu, and Miodrag Potkonjak

Keyless Public Watermarking for Intellectual Property Authentication 96
Gang Qu

Efficiency Improvements of the Private Message Service 112
Oliver Berthold, Sebastian Clauß, Stefan Köpsell, and Andreas Pfitzmann

A Reputation System to Increase MIX-Net Reliability 126
Roger Dingledine, Michael J. Freedman, David Hopwood,
and David Molnar

Neural Networks Functions for Public Key Watermarking 142
Justin Picard and Arnaud Robert

A Graph Theoretic Approach to Software Watermarking 157
Ramarathnam Venkatesan, Vijay Vazirani, and Saurabh Sinha

COiN-Video: A Model for the Dissemination
of Copyrighted Video Streams over Open Networks 169
Dimitris Thanos

Natural Language Watermarking: Design, Analysis, and
a Proof-of-Concept Implementation 185
Mikhail J. Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann,
Florian Kerschbaum, Dina Mohamed, and Sanket Naik

Digital Watermarking of Chemical Structure Sets 200
Joachim J. Eggers, Wolf-Dietrich Ihlenfeldt, and Bernd Girod

The Strong Eternity Service 215
Tonda Beneš

Real World Patterns of Failure in Anonymity Systems 230
Richard Clayton, George Danezis, and Markus G. Kuhn

Traffic Analysis Attacks and Trade-Offs
in Anonymity Providing Systems 245
Adam Back, Ulf Möller, and Anton Stiglic

Automatic Detection of a Watermarked Document Using a Private Key ... 258
Julien P. Stern and Jean-Pierre Tillich

Zero-Knowledge Watermark Detection and Proof of Ownership 273
André Adelsbach and Ahmad-Reza Sadeghi

F5—A Steganographic Algorithm:
High Capacity Despite Better Steganalysis 289
Andreas Westfeld

A Collusion-Secure Fingerprinting Code Reduced
by Chinese Remaindering and Its Random-Error Resilience 303
Hirofumi Muratani

Practical Capacity of Digital Watermarks 316
Ryo Sugihara

Blur/Deblur Attack against Document Protection Systems Based
on Digital Watermarking 330
Sviatoslav Voloshynovskiy, Alexander Herrigel, and Thierry Pun

Second Generation Benchmarking and Application Oriented Evaluation ... 340
Shelby Pereira, Sviatoslav Voloshynovskiy, Maribel Madueno,
Stéphan Marchand-Maillet, and Thierry Pun

Robust Covert Communication over a Public Audio Channel
Using Spread Spectrum 354
Darko Kirovski and Henrique Malvar

Hiding Association Rules by Using Confidence and Support 369
Elena Dasseni, Vassilios S. Verykios, Ahmed K. Elmagarmid,
and Elisa Bertino

How to Break a Semi-anonymous Fingerprinting Scheme 384
Ahmad-Reza Sadeghi

An Analysis of One of the SDMI Candidates 395
Julien Boeuf and Julien P. Stern

Author Index 411

Trustworthy Paper Documents

Marshall Bern, Jeff Breidenbach, and David Goldberg

Xerox Palo Alto Research Center
3333 Coyote Hill Rd., Palo Alto, CA 94304, USA
{bern,jbreiden,goldberg}@parc.xerox.com

Abstract. In the first part of this paper, we propose a watermarking method for text documents that is less vulnerable to attacks than previous methods. In the second part, we propose a method for protecting the actual content of the document. In a section of independent interest, we discuss the interplay between error-correcting codes and steganography.

1 Introduction

How do we know whether to trust a paper document? For hundreds of years, people have relied on handwritten signatures, along with special inks, seals, and papers, to guard against forgery. We would like to invent equivalents of these traditional techniques for computer-based word processing and typesetting.

How can we prevent unauthorized copying of a paper document? It is essentially impossible to prevent an adversary from copying, so the goal here is to deter copying by the promise to discover it after the fact. The suggested method of accomplishing this goal is to embed a hard-to-remove identifier into each copy of the document, but so far no method of embedding such an identifier has proved resistent to attacks.

Before we can consider the problems of forgery and copy deterrence, we need to fix some concepts and terminology. We use the term *fingerprint* to mean a hidden, hard-to-remove identifier that encodes the name of the recipient of a document. A fingerprint enables *traitor tracing*, that is, it provides a way to discover the source of unauthorized copying. We use the term *watermark* to mean a visible, hard-to-remove identifier that encodes the name of the author or owner of a document. A watermark asserts ownership and deters copying.[1] A watermark that is hard to make serves another purpose as well. Such a watermark gives some assurance of the *authenticity* of the document, the fact that the document did indeed originate with the claimed author. Document *integrity* goes beyond authenticity: integrity means that no tampering with the content has occurred since authorship. Finally, *steganography* is any sort of covert writing, in which not only the message but also the location of the message is secret.

In this paper we present two different approaches to trustworthy paper documents. The first approach emulates traditional methods such as notary's stamps and seals, in order to provide traitor tracing along with a weak guarantee of authenticity. This approach uses fingerprints and watermarks so intertwined with the text that they would

[1] What we call a fingerprint is a "robust invisible watermark" for Mintzer el al. [15], and what we call a watermark is a "robust visible watermark".

I. S. Moskowitz (Ed.): IH 2001, LNCS 2137, pp. 1–12, 2001.

be hard for an adversary to remove or copy even knowing that they are there. Previous fingerprinting methods for text, reviewed below, are prone to washing attacks.

The second approach transfers modern cryptographic methods such as digital signatures to paper. This approach has the advantage of offering a strong guarantee of authenticity and integrity, but the disadvantage that it requires a larger change in current practice. The second approach deters copying of a digitally signed document, but it does not deter copying of the human-readable text, in fact, it makes fingerprinting of the text more difficult.

2 Previous Work

The literature on hiding data in images is large and rapidly growing; the three workshops on information hiding [1,4,18] give a good overview of the field.

Text document images, however, are quite special types of images, which have large blank areas, structured frequency spectra, and small meaningful subunits (words and letters). Generic image fingerprinting schemes are not applicable to text images, at least not without major modifications. For example, NEC's well-known spread-spectrum method [9] adds a fingerprint in the frequency domain by making small changes to the signal-carrying frequency bins. Such a fingerprint applied to a text image, however, introduces ghostly squiggles in the white space and is hence quite visible. (This fingerprint, whether used on a generic or a text image, is also fairly easily removable by a small non-linear distortion of the image [17].)

Several researchers have considered the special case of hiding data in text images. Brassil and colleagues proposed modulating interline [5] and/or interword[2] spacing [6,14]; a more recent proposal [7] vertically shifts words relative to the baseline of the line. Groups at Xerox PARC have experimented with modulating serif lengths and heights and shapes of letters.

Each of the methods just mentioned, however, is prone to attack. A relatively simple piece of software—essentially the same as the program that reads the hidden data—can find lines, words, or letters within a document image and modify them appropriately in order to remove or counterfeit data. For some steganographic methods, the attacking software already exists: passing the document image through a token-based compressor[3] with xy-coordinates quantized to $1/300$ inch would *wash off* most of the vertical shifts of $1/600$ inch proposed by Brassil and O'Gorman [7].

General-purpose steganography, however, is a harder problem than hiding a fingerprint. The difference is that a fingerprint need only be readable by its author (and perhaps later by a court of law), whereas a steganographic message must be readable by a recipient other than the author. We exploit this difference in our first approach to paper document security.

[2] Interword spacing had perhaps been used before. Anderson [2] repeats a story that in the 1980's, Margaret Thatcher's government, fed up with leaks of cabinet documents, reprogrammed the word processors to add a white-space fingerprint to documents they produced.

[3] Token-based compression [3,12,13,19] encodes a text document by a set of representative bitmaps—letters and symbols—and the xy-coordinates of each appearance of a representative.

3 First Approach

The key idea in our first approach is a countermeasure to the washing attack: background marks that make automatic segmentation and washing of letters—but not human reading—quite difficult. Such background marks are feasible for text documents, because human reading abilities far surpass the capabilities of machine optical character recognition (OCR). Our countermeasure is intended to foil automatic washing attacks; our methods cannot guard against time-consuming manual attacks such as completely retyping the document.

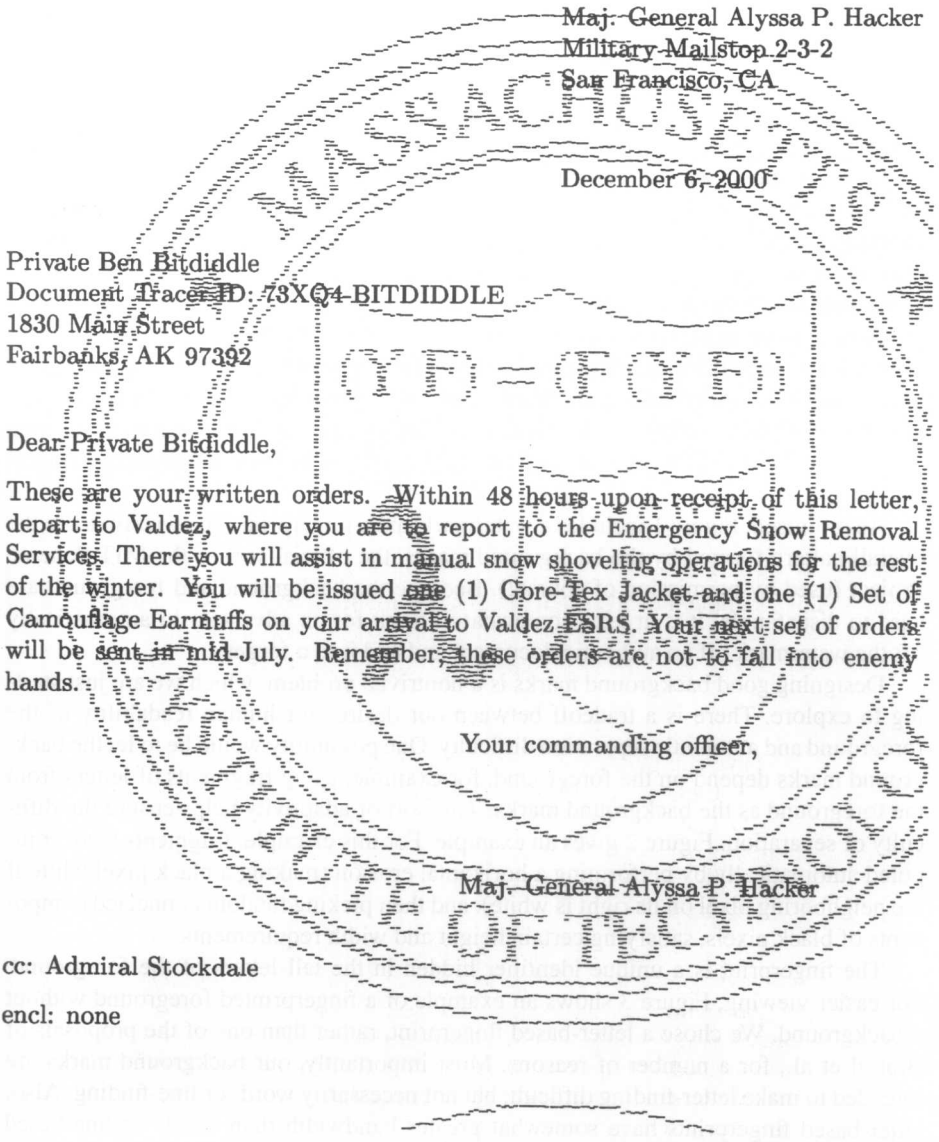
Figure 1 shows an example. The background marks are black rectangles, measuring 2×8 pixels, with slightly randomized locations. Imagine a washing program that attempts to separate the background marks from the foreground text. If the program removes only clean, well-defined background marks, it will leave a menagerie of hairy letters that would be difficult to repair or wash automatically. On the other hand, if the program is more aggressive and removes all possible background marks, it will leave broken letters, equally difficult to process. Repairing or washing letters is approximately as hard as OCR, which fails quite dramatically with severely connected or broken letters. To defeat the washing attack, it is not necessary that every letter be hard to separate from the background, but only enough letters (say as few as 20 or 30) to carry a fingerprint.

As promised, Figure 1 includes a watermark and a fingerprint. The watermark is the overall pattern formed by the background marks, the official seal of the MIT Scheme Project (used by permission of course!). Because the background and foreground are hard to disentangle, it is at least somewhat difficult for an adversary to automatically lift the watermark off an authentic document and add it to a forgery.

Designing good background marks is a nontrivial problem, which we are just starting to explore. There is a tradeoff between our desires for human readability of the foreground and automatic separation difficulty. One possibility would be to let the background marks depend on the foreground, for example, using fragments of letters from the foreground as the background marks. This sort of adaptivity helps ensure the difficulty of separation. Figure 2 gives an example. For this example, fragments were computed automatically by performing a horizontal erosion (making a black pixel white if the neighboring pixel on its right is white), and then picking random connected components of black pixels, satisfying certain height and width requirements.

The fingerprint is a unique identifier hidden in the tall letters of the foreground. For easier viewing, Figure 3 shows an example of a fingerprinted foreground without a background. We chose a letter-based fingerprint, rather than one of the proposals of Brassil et al., for a number of reasons. Most importantly, our background marks are intended to make letter-finding difficult, but not necessarily word- or line-finding. Also, letter-based fingerprints have somewhat greater bandwidth than word- or line-based fingerprints. Finally, modifications to the letters should be slightly harder to wash off, even without the hard-to-separate background.

We now give the details of the fingerprinting program, starting with how the program decides what is a tall letter. The program starts from a raster document image, which may be either a scan or an original. It performs a horizontal dilation in order to find lines on the page, and computes a nominal baseline location for each line.



Maj. General Alyssa P. Hacker
Military Mailstop 2-3-2
San Francisco, CA

December 6, 2000

Private Ben Bitdiddle
Document Tracer ID: 73XQ4-BITDIDDLE
1830 Main Street
Fairbanks, AK 97392

Dear Private Bitdiddle,

These are your written orders. Within 48 hours upon receipt of this letter, depart to Valdez, where you are to report to the Emergency Snow Removal Services. There you will assist in manual snow shoveling operations for the rest of the winter. You will be issued one (1) Gore-Tex Jacket and one (1) Set of Camouflage Earmuffs on your arrival to Valdez ESRS. Your next set of orders will be sent in mid-July. Remember, these orders are not to fall into enemy hands.

Your commanding officer,

Maj. General Alyssa P. Hacker

cc: Admiral Stockdale

encl: none

Fig. 1. A document with a hidden fingerprint and a visible watermark. The fingerprint is written by stretching and shrinking tall foreground letters

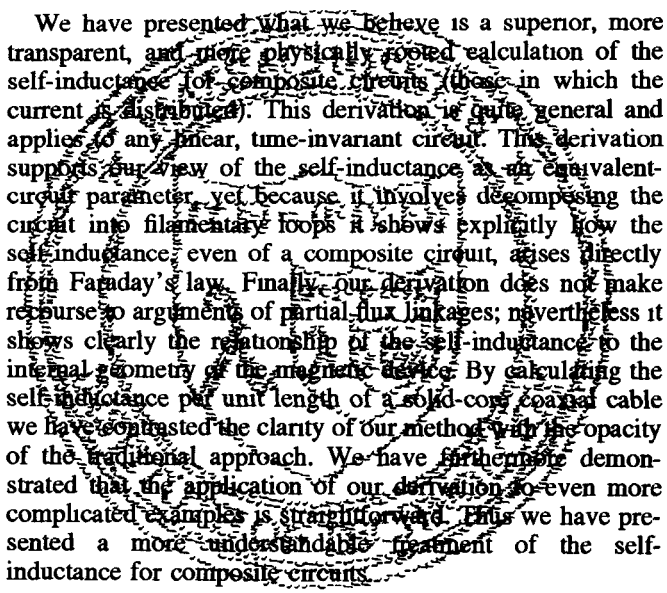


Fig. 2. In this example the background marks are random fragments of the foreground text

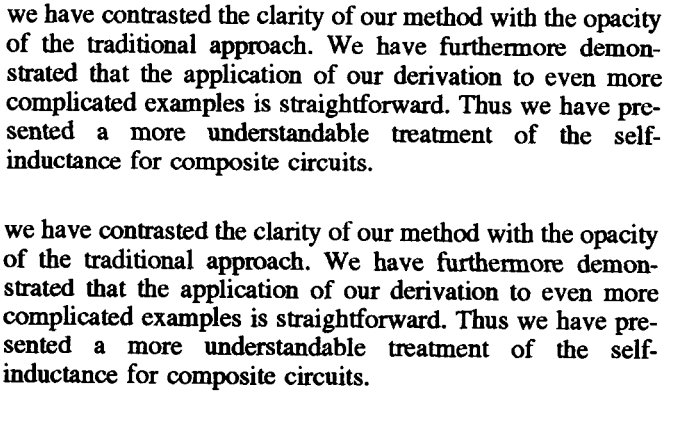


Fig. 3. The upper piece of text is unmodified. The lower piece contains a fingerprint written with stretched and shrunk tall letters. In the next-to-last line the **d** in **sented** has shrunk, and in the last line both the **d** in **inductance** and the **f** in **for** have grown

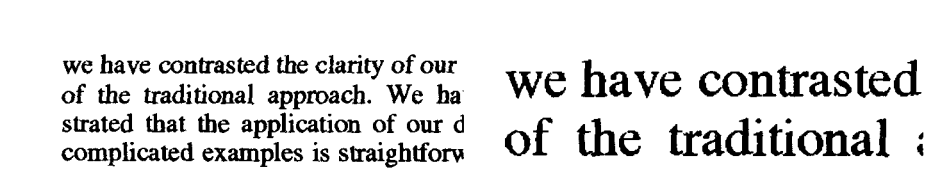


Fig. 4. (a) Example text with a fingerprint consisting of random bumps and bites around the perimeters of letters. (b) Magnified by a factor of two

It then computes connected components (cc's for short) of black pixels. The most frequent height of a cc (that is, the mode in a histogram of heights) is assumed to be the standard height of a lower-case letter—the *x-height* in font designer's terminology. The most frequent height of a cc that is between 1.2 and 2.0 times the x-height is defined to be the standard height of a tall letter. Each cc at most 1 pixel taller or shorter than this standard height, with a baseline within 1 pixel of the baseline of the line containing the cc, is defined to be a *tall letter*. For most roman fonts, tall letters include **H**, **h**, and **k** but not **t** or **p**.

To stretch a tall letter and write a 1, the program duplicates a row of pixels halfway between the x-height and the tall-letter height. To shrink a tall letter and write a 0, the program removes that row of pixels. The exact choice of row is not critical, so long as it is not too close to the tall-letter height, where it could change the cross-bar in a **T** or the serifs in an **H**, nor too close to the x-height where it could change the middle stroke in an **H** or **E**.

For the fingerprint, there is a tradeoff between our desires for high machine readability and low human perceptibility. We have found that shrinking and stretching tall letters by 1/300 inch is reliably readable in a 300 dpi binary scan, even after severe degradation by multigeneration copying. In fact the hidden data may be more robust than the original text! After seven generations of copying with various darkness settings, the text was no longer legible yet about 75% of the hidden bits were read correctly. Modifications of 1/300 inch, however, are slightly obtrusive to the eye. They become less obtrusive if all tall letters in each word are modified in the same way. For still lower perceptibility, we can restrict which tall letters carry data, for example, using only occasional tall letters chosen by a pseudorandom number generator.

We have also experimented with another fingerprinting method, in which a few bits are added or subtracted from a random location around the perimeter of a letter in order to write a bit. See Figure 4. In this implementation, we randomly pick two adjacent rows or columns and randomly pick either "first" or "last". Then the first or last black pixel in each of these rows or columns is turned white to write a 0; the first or last white pixel in each of these rows or columns is turned black to write a 1. The overall effect is something like scanner or printer noise.

When combined with a hard-to-separate background, both the tall-letter and perimeter fingerprints should be fairly resistant to automatic washing attacks, such as those based on OCR. The perimeter fingerprint may be a little more resistant than the tall-letter fingerprint, because there are many more possible hidden bit locations. By writing the fingerprint into only a small fraction of the possible bit locations, we gain resistance to *collusion attacks*, which combine a number of copies of the same document. The

average of a small number of copies is then likely to contain traces of all the fingerprints. Another technique to help foil collusion attacks is to vary the background marks by a few bits from copy to copy. If this technique were not used, attackers could take advantage of the fact that background marks were fixed but fingerprint-carrying foreground letters were variable.

Although finding the fingerprint-carrying letters is hard for an adversary, this task is relatively easy for the originator of the document, who has access to original images of background and foreground. A sort of local warping algorithm [22] can match even a degraded multigeneration copy with the original images within a tolerance of one to two pixels. The originator uses this algorithm to read the fingerprint on a recovered pirate copy of the document. Each data-carrying letter on the pirate copy is compared with the corresponding letter on the original unmodified text image. For legal purposes, the originator or a trusted third party should keep a reliably dated copy of the original text image and the fingerprinting program, in order to prove that the fingerprint was read honestly rather than fabricated after the fact.

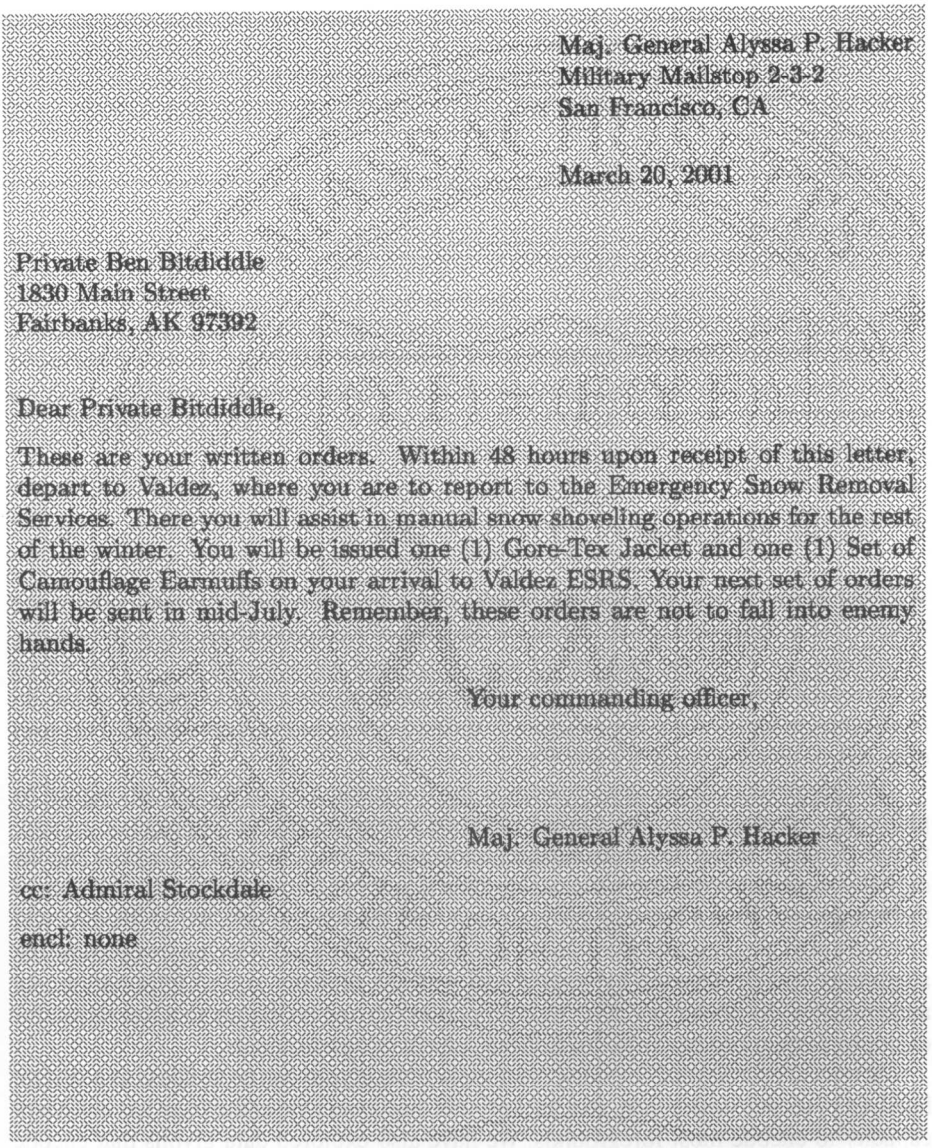
4 Second Approach

We now move on to our second approach, a way to ensure both the authenticity and integrity of a paper document. We reuse the idea of background marks, with two major differences. First, the background and foreground are designed to be more easily machine-separable. Second, the background marks are *data glyphs*, marks which directly encode binary data. For example, a method developed at Xerox PARC [11] uses diagonal slashes to encode 0's and 1's.

The glyphs encode the same content as the foreground, but as digital data signed by public key cryptography. The content could be a page descriptor language or a (compressed) image. To verify the document's authenticity, the document is scanned and the digitally signed information in the background is automatically compared to the scan of the foreground. If an adversary has tampered with the background, the glyph data will not decrypt to meaningful information; and if he has tampered with the foreground, the alterations will be detected by comparison with the digitally signed information. An authenticated document has other nice properties beyond resistance to adversaries; for example, perfect copies can be made from stained or torn originals.

At typical printing and scanning resolutions, data glyphs can be as dense as a several hundred bytes per square inch, so a background that covers the entire page has sufficient capacity to encode most foregrounds. We have also explored data glyphs that encode only a cryptographically secure "checksum" [20], rather than the entire foreground. This idea holds the promise of reducing the background data to a small notary's stamp, albeit at the cost of losing some of the other nice properties.

Figure 5 shows an example of an authenticated paper document. Notice that the background marks here are smaller and less obtrusive than the background marks in Figure 1. They are still less obtrusive when printed in a different color than the foreground, but we could not show this here. The marks are written all over the page in a uniform grid; glyphs that are obscured by foreground text either carry no data or are


Maj. General Alyssa P. Hacker
Military Mailstop 2-3-2
San Francisco, CA

March 20, 2001

Private Ben Bitdiddle
1830 Main Street
Fairbanks, AK 97392

Dear Private Bitdiddle,

These are your written orders. Within 48 hours upon receipt of this letter, depart to Valdez, where you are to report to the Emergency Snow Removal Services. There you will assist in manual snow shoveling operations for the rest of the winter. You will be issued one (1) Gore-Tex Jacket and one (1) Set of Camouflage Earmuffs on your arrival to Valdez ESRS. Your next set of orders will be sent in mid-July. Remember, these orders are not to fall into enemy hands.

Your commanding officer,

Maj. General Alyssa P. Hacker

cc: Admiral Stockdale

encl: none

Fig. 5. An authenticated document. The background data glyphs encode a digitally signed copy of the foreground text

fixed by an error-correcting code. In this case, the background seal is mere decoration: the security of the document does not depend upon the difficulty of removing the seal.

If a public key infrastructure (PKI) for printers and document verifiers was deployed, and if users of paper could be trained to trust only documents that pass a verification test, then authenticated paper documents would offer a strong solution to security

problems. We would trust authenticated documents, because the background could not be forged without access to a private key. We would be able to trace a traitor who passed out authenticated documents, because the foreground and background would include a unique identifier, the removal of which would cause the document to fail the verification test. Paper would inherit the security capabilities of digital media, while keeping most of its current affordances such as ubiquity, durability, and portability.

Of course, current practice is quite far from this scenario. PKI's are still fairly rare. More problematically, this scenario does not prevent copying of unauthenticated foregrounds. Thus, even in a digital future, our second approach may not be appropriate for applications such as tracing leaks to the press.

When we try to combine our two approaches, we run into some difficulties. We can tuck a fingerprint into the foreground letters just as before. But now we have competing requirements for background marks: they should be hard to separate to protect against washing attacks, and yet easy to read in order to carry the authentication data. A still trickier problem is that we cannot let the authentication data make the fingerprint irrelevant or washable. For example, if the background encodes text without a fingerprint, then a traitor could pass along just the decrypted background and avoid identification. If the background contains the image of the fingerprinted foreground text, then a traitor could use this image to separate foreground from background and proceed with a washing attack. A background encoding only a checksum is one possible solution to this problem. An alternative is a background that contains a fingerprinted text image, formatted quite differently from the foreground text and containing its own hard-to-separate seal. This alternative, however, would require manual rather than automatic verification.

5 Error-Correcting Codes and Steganography

Almost any sort of steganography benefits from the use of error-correcting codes. In this section we discuss some peculiar properties of steganography that affect error correction.

The basic channel coding model assumes that bits are misread with certain probabilities. In steganography, however, it is also quite common that a bit is not read at all or a false bit is read when no bit was sent. These *synchronization errors* are common, because in steganography not only may it be difficult to read a bit, but it may even be difficult to recognize the hiding places for bits.

Our fingerprinting application is not a good illustration of this phenomenom. In this application the sender and receiver are the same person, and he can simply recall the hiding places rather than recompute them. But imagine that two spies attempt to stretch and shrink tall letters in order to communicate with each other through innocent-looking paper documents. Now the receiver must recompute which cc's qualify as tall letters, and will inevitably come up with a slightly different set of cc's.

Thus an error-correcting code for this steganographic application must be one that can cope with frequent synchronization errors. For this reason, we think that convolutional codes are more suitable than classical block codes. Block codes handle desynchronization by reframing the data in a sort of exhaustive search, whereas convolutional

codes can gracefully incorporate synchronization errors into a Viterbi (maximum likelihood) decoding [16]. Since steganographic messages are usually quite short and decoding is not a time-critical operation, Viterbi decoding should be quite practical for steganography.

Another peculiarity of steganography is that some bits are more likely to be corrupted than others. This situation occurs because not all hiding places are equally good. For example, in our fingerprinting method a foreground letter may obscure a background glyph or a background glyph may obscure the height of a tall letter. Similarly the interword spacing is harder to measure in **some where** than in **all here**, and the baseline is harder to locate in **egg** than in **nominal**.

The sender has some idea of the error rate for each data bit at the time that he hides it. In some situations, for example when the sender and the receiver are the same person, the receiver also has an estimate of the error rate and can incorporate these probabilities into the Viterbi decoding. In other situations, such as the spy example above, the receiver cannot easily estimate the probability of an error: he cannot tell if **X** is unusually short because it was shrunk or because this rare letter is always a bit shorter in the given font. What can the sender do in the case that he knows the error rate but the receiver does not?

The question of channel coding with side information at the sender has already been considered in the coding literature, without explicit mention of the steganography connection. In a classic paper, Shannon [21] showed that, asymptotically, for a binary channel, the sender cannot make use of his knowledge of the corruptibility of bits: he may as well use a general error-correcting code designed for a uniform error rate. In a more recent paper, Costa [8] showed that for a real-valued channel, in which the side information is a pre-existing Gaussian signal, the sender can make very good use of the side information, in effect canceling the pre-existing signal.

There is a striking contrast between these two results: Shannon's result is bad news, asymptotically the worst possible, whereas Costa's result is good news, asymptotically the best possible! Which model applies to our situation? Textual steganography is "nearly binary", because text images are black and white, and because letter shapes and locations cannot be varied by more than one or two pixels without becoming easily perceptible. Hence we believe that Shannon's model is more relevant than Costa's model. However, other steganographic situations (see [10]) are closer to Costa's model.

6 Conclusions

Our first proposal adds another text fingerprinting and watermarking scheme to the stack of already existing schemes, advancing the "arms race" between watermarkers and attackers. Our main innovation is the exploitation of a human advantage over machines: our ability to read noisy text. We were somewhat surprised to find that the combined problem of fingerprinting and watermarking seems to be more tractable than fingerprinting alone.

Our second proposal transfers modern cryptographic methods such as digital techniques to paper. The security properties of digital techniques are relatively well known, but the interaction of these properties with human work practice—especially such long established practice as use of paper documents—is still unclear.