Quantifying Information Leakage in Document Redaction

Daniel Lopresti
Computer Science & Engineering
Lehigh University
Bethlehem, PA 18015, USA
lopresti@cse.lehigh.edu

A. Lawrence Spitz
DocRec Ltd
34 Strathaven Place
Atawhai, Nelson, New Zealand
spitz@docrec.com

ABSTRACT

In this paper, we examine ways in which sensitive information might leak through the process of redaction. Such attacks apply known methods from document image analysis and natural language processing to recover text thought to have been obliterated for the purposes of public release. Systematically identifying and testing these weaknesses is a first step towards designing effective countermeasures. We describe our development of a prototype semi-automated system intended to accept as input a redacted document and provide feedback to the user as to whether the document might suffer from such leaks.

Categories and Subject Descriptors

I.7.5 [**Document and Text Processing**]: Document Capture—*document analysis*

General Terms

algorithms, measurement, security

Keywords

character shape coding, classified documents, document analysis, document redaction

1. INTRODUCTION

It has been recently demonstrated, in dramatic fashion, that sensitive information thought to be obliterated through the process of redaction[1] can be successfully recovered via a combination of manual effort, document image analysis, and natural language processing (NLP) techniques. As reported in international news media [2], computer security researchers David Naccache and Claire Whelan demonstrated that they could recover two instances of redacted text from images of previously classified U.S. intelligence memos by measuring font design attributes and matching them against a font database. An image of one of the documents is shown in Figure 1, where they were able to determine that the missing word in the second case was almost certainly "Egyptian."

In retrospect, this achievement should come as no surprise to those versed in the field of document analysis, as efforts to recognize naturally degraded text have led to numerous research results that correspond almost exactly to some of the steps used by Naccache and Whelan. Even so, this appears to be a problem worthy of study for several reasons. Clearly, the potential exists for the unintended release of sensitive information which could be disastrous. A full and thorough understanding of what is technically possible from an adversarial standpoint can lead to better methods, and perhaps even automated or semi-automated tools, to guarantee that hidden text remains hidden. At the very least, the problem brings together aspects of document analysis research in somewhat different ways from traditional applications. Hence, it may offer the opportunity for new insights with broader implications.

So far as we know, there are no previous publications examining this issue from a research standpoint. Naccache and Whelan are currently in the process of preparing a paper describing their result, which was presented informally at a European computer security conference [9]. In a paper now under submission [8], we outlined in some detail what might be revealed through redaction, exploring how known methods from document image analysis and NLP might be employed to detect vestigial artifacts of the pre-redacted text. We also presented preliminary experimental results and described a series of increasingly stringent countermeasures to address, and in some cases eliminate, the perceived threat.

In the present paper, we outline the initial development of a semi-automated system intended to accept as input a redacted document and provide feedback to the user, presumably an individual charged with clearing the material for public release, by producing one of two outputs:

1. PASS: this document is okay to release as-is; it does not appear to reveal sensitive information.

2. FAIL: this document must not be released without further redaction because it is susceptible to one or more known types of attack.

Ideally, this determination will be accompanied by a detailed analysis of how information might leak under different scenarios. While no such system can be perfect, we believe

[1]Redaction is defined as "the removal of exempted information from copies of a document" [7].

HDP'04, November 12, 2004, Washington, DC, USA.

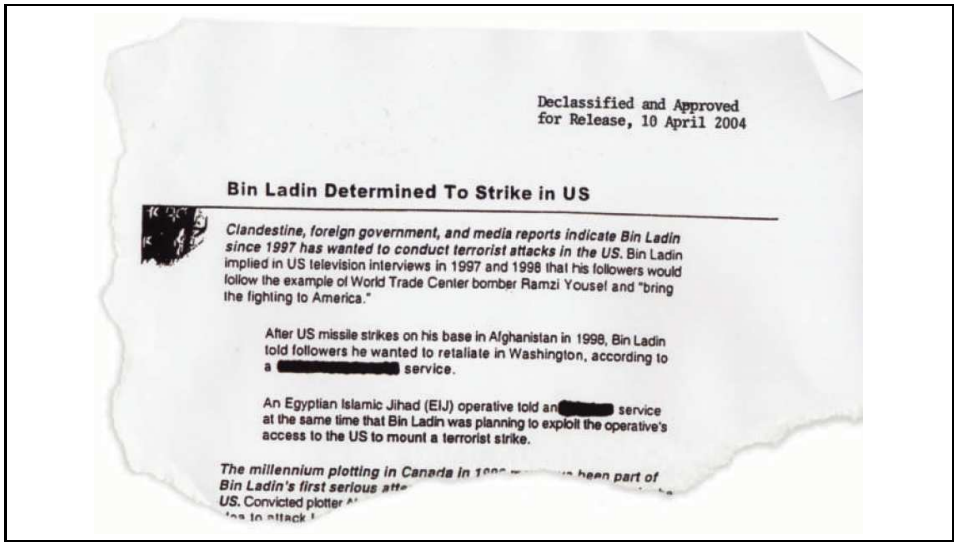

Figure 1: Part of the redacted document decoded by Naccache and Whelan (from [2]).

it could provide valuable assistance to humans who must perform the task of redaction.

2. INFORMATION LEAKAGE

Typically, redaction is accomplished by obscuring the text in question using a thick black marker pen, covering up the text with a special opaque tape, or excising a section from the document using, for example, a razor knife. In all cases, the physical redaction takes place on a copy of the original document, and a copy of the redacted document is what gets released to the public.

If an adversary can determine with some level of confidence the identity of one or more of the redacted regions on a page, we consider it a serious breach. We note that there are differing degrees of severity. In the worst case, the attacker is able to determine with high certainty precisely one interpretation for the given redaction. Under a somewhat less serious scenario, he or she might be able to narrow down the options to a small number of choices (say, two to ten). We can quantify what the attacker is able to glean in terms of orders of magnitudes: tens, hundreds, thousands of possibilities, etc. At some point, the guesses become essentially random and the document is considered to have been securely redacted.

How can information leak when text has been redacted? We enumerate some potential ways this might happen:

1. The text has not been completely obliterated (*e.g.*, the reflective qualities of "black" may differ for laserprinter toner and a marker pen).

2. Although largely obscured, certain features of the text may remain apparent (*e.g.*, the numbers and locations of ascender and descender characters). As Spitz has shown in his work on character shape coding [13, 15, 14], this is often enough to narrow down the set of choices to a small number.

3. When using a monospaced font (such as Courier), the width of the gulf between the cleartext immediately adjacent to the redacted region reveals the length, in characters, of the missing word(s). Combined with language modeling techniques, this may be sufficient information to reveal the missing text, or at least limit the possibilities.

4. For proportionally-spaced fonts, based on the locations of adjacent words and estimates of character widths (which are easy to obtain through a number of methods, including an analysis of cleartext on the page in question), the widths of missing words can be estimated and, as in the monospaced case, NLP brought to bear.

In many cases, the above attacks will not be sufficient to identify the missing text uniquely. Moreover, except perhaps for the first and second attacks listed, none will work when large regions of the page are redacted. Keep in mind, though, that the performance measure here is quite different from a traditional pattern recognition problem. In the latter case, we might demand 95% or 98% accuracy before we declare success. In a security application such as the one we are considering, however, if an adversary is able to recover redacted material even 2% of the time, that qualifies as a success for the attacker and a failure for the process, with perhaps serious consequences.

3. TECHNIQUES USED FOR MOUNTING ATTACKS ON REDACTED TEXT

In this section, we enumerate some of the approaches that might be used by an adversary in attempting to recover redacted text.

3.1 Image Processing

Redacted documents are most often seen as photocopies of a redacted "original." It is usual in the document recognition community to deal in bitonal (one bit per pixel) images of documents wherever possible. Processing of documents redacted using felt-tipped pens, however, seem to require that at least preliminary image processing be done in the grayscale domain.

For example, taking the image shown in Figure 2, it can be seen that the recovery of the redacted word is relatively straightforward using known image processing techniques.

3.2 Font Metrics

The work of Naccache and Whelan, as described by Butler [2], relied on actual font identification based on minimization of the Hamming distance between noise-free renderings of the characters found in the scanned images and the images themselves. They apparently relied on high quality rendition such that the font metrics were not sufficiently distorted as to make identification impossible, and therefore selection of candidate words much more difficult.

Khoubyari and Hull [5] have also performed predominant font identification. Kopec [6], Zramdini and Ingold [18], Fang and Hull [3], and others have shown it is possible to perform font identification even from somewhat degraded images, and we posit here that this step may not even be necessary because character set width information can be obtained from the unredacted content on the same page.

Hull, *et al.* [4] went on to identify whole words in images artificially degraded to simulate facsimile scanning and printing.

3.3 Taking Advantage of Artifacts

Because the obscuration of word images is an inexact process, there may be other artifacts available to use besides word set widths. Ascenders, descenders, or i-dots that have not been occluded in the redaction process may be visible. Noting the presence of such artifacts and, better yet, their approximate position within the word can materially reduce the candidate list. Moreover, this can also be used to improve font metric determination. Even the absence of such artifacts, if it can be detected, is useful to an attacker.

3.4 Natural Language Processing

If the redacted sentence can be parsed (if it is, in fact, a sentence), the number of candidate words can be reduced significantly by using a part-of-speech tagged lexicon, only selecting those words that would preserve the parsability of the sentence.

Proper nouns are among the most redacted tokens. Comprehensive lexica of proper nouns are difficult to assemble and maintain, although online resources, including the growing pervasiveness of commercial search engines such as Google, may have the effect of making this task easier than before. Probably the best strategy for an attacker is to develop and maintain local lexica of potentially redact-prone terms. Number strings also sometimes represent sensitive information and may require a fundamentally different approach.

4. INITIAL EXPERIMENTAL RESULTS

In this section, we describe the results of some preliminary experiments we have started to perform. We begin by noting that due to the unusual nature of the redaction problem, we are limited to using synthetic data in all of our work. We can do no better than to hypothesize about the kinds of things that might go wrong and how an attacker might try to exploit them.

Based on the work reported by Naccache and Whelan, it would appear that an important channel for information leakage is an estimate of the width of the redacted text in combination with language statistics. To study this question, we conducted a simple experiment that involved computing the widths of text strings chosen from several large lexica and determining the degree to which this one simple feature revealed information about the text itself. We conducted this test for three fonts, Times, Helvetica, and Courier, using character width and typesetting information from the Adobe font metrics file [1], and assuming a 12-point typeface and scanning at 600 dpi.

We considered four lexica:

YAWL "yet another word list" is a free, public domain list of over 264,000 English words [17].

COUNTRIES is a comprehensive list of 416 country names from around the world, including both official ("Republic of Indonesia") and informal ("Indonesia") variants.

CONGRESS is a list of the full names of the 101 Senators (including the Vice President, who presides over the Senate) and the 439 Representatives currently serving in the U.S. Congress.

NAMES is the cross-product of two lists of names provided by the U.S. Census Bureau [10]. The first is a merged list of male (1,219) and female (4,275) first names, while the second is a list of last names (88,799). By enumerating all possible {first name, last name} pairs, a total of 487,861,706 names are generated. Many of these obviously do not correspond to real people because the current U.S. population is only 294,193,053 as of this writing [12]. Still, we can be certain that our exhaustive enumeration covers roughly 90% of the U.S. population since that is the lower bound on the coverage for each of the individual lists we used.

In each case, we computed the average number of text strings of a given width when typeset in the font in question (see Table 1). This can be taken as a measure of the overall uniqueness of information contained in redacted regions under various scenarios. If this number is low, then it would be easy, on average, for an attacker to determine the missing text based solely on the width of the redaction.

Table 1: Average number of strings of a given width.

	Font		
Lexicon (Size)	*Times*	*Helvetica*	*Courier*
YAWL (264,057)	290	261	548
COUNTRIES (416)	1.33	1.28	1.97
CONGRESS (540)	1.66	1.60	2.90
NAMES (487,861,706)	481,125	427,573	969,903

As can be seen from this analysis, certain classes of data are simply unsafe; if it is known that the redacted text is the name of a country or a member of Congress, the adversary can, with high probability, recover it. It is also interesting to note that, from this standpoint, Courier is a safer font to use than either Times or Helvetica. This is because it conveys less information since all of its characters are the same width.

Moving beyond the average case, we believe it is also important to consider worst-case analyses. As noted previously, security applications are unlike most traditional pat-

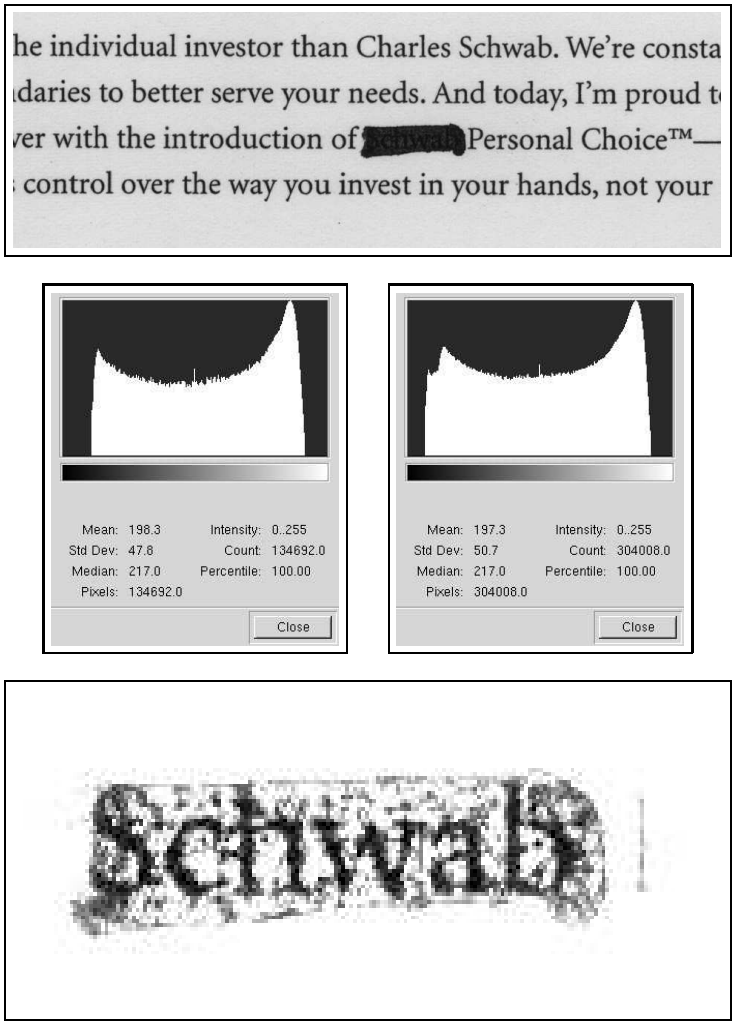

Figure 2: A portion of a "redacted" page. From top to bottom we have the source image; histograms of (a) the background consisting of portions of the image not including the redacted word, and (b) the entire redacted image showing the extension of the dark portion of the histogram; and the redacted region after the histogram analysis yields threshold values.

tern recognition problems. If it is critical to avoid any leakage of information through redaction whatsoever, it may be desirable to develop special-purpose tools for checking redacted material to certify that the worst case has indeed not arisen.

In Table 2, we show the number of text strings which share their widths with at most two other text strings. These are the kinds of redactions that would be easy for an adversary to reverse-engineer; note that every lexicon-font combination possesses some number of such cases.

Table 2: Worst-case analysis of information leakage through text string widths.

		Font	
Lexicon (Size)	*Times*	*Helvetica*	*Courier*
YAWL (264,057)	213	234	112
COUNTRIES (416)	394	393	260
CONGRESS (540)	472	514	233
NAMES (487,861,706)	49	78	20

5. PLUMBER: A TOOL FOR DETECTING INFORMATION LEAKAGE

We have already begun to build a prototype system to test these ideas; a screen snapshot is shown in Figure 3. Plumber is written in Tcl/Tk, a general-purpose scripting language that is popular for implementing user interfaces [11, 16]. As demonstrated in the figure, Plumber integrates a graphical browser for scanned page images along with document analysis and language resources useful for checking for leaks. The system is semi-automated; the user interacts with the page in question, for example, by marking it up to delineate redacted regions, spacing widths, etc. and designating the suspected font, and then invokes various functions for confirming whether the information that is intended to be hidden might be revealed.

When a potential decoding is determined, Plumber allows the user to render the string in the indicated font and overlay it on the redacted region (Figure 4). In this way, the user can confirm whether or not the string fits based on the other

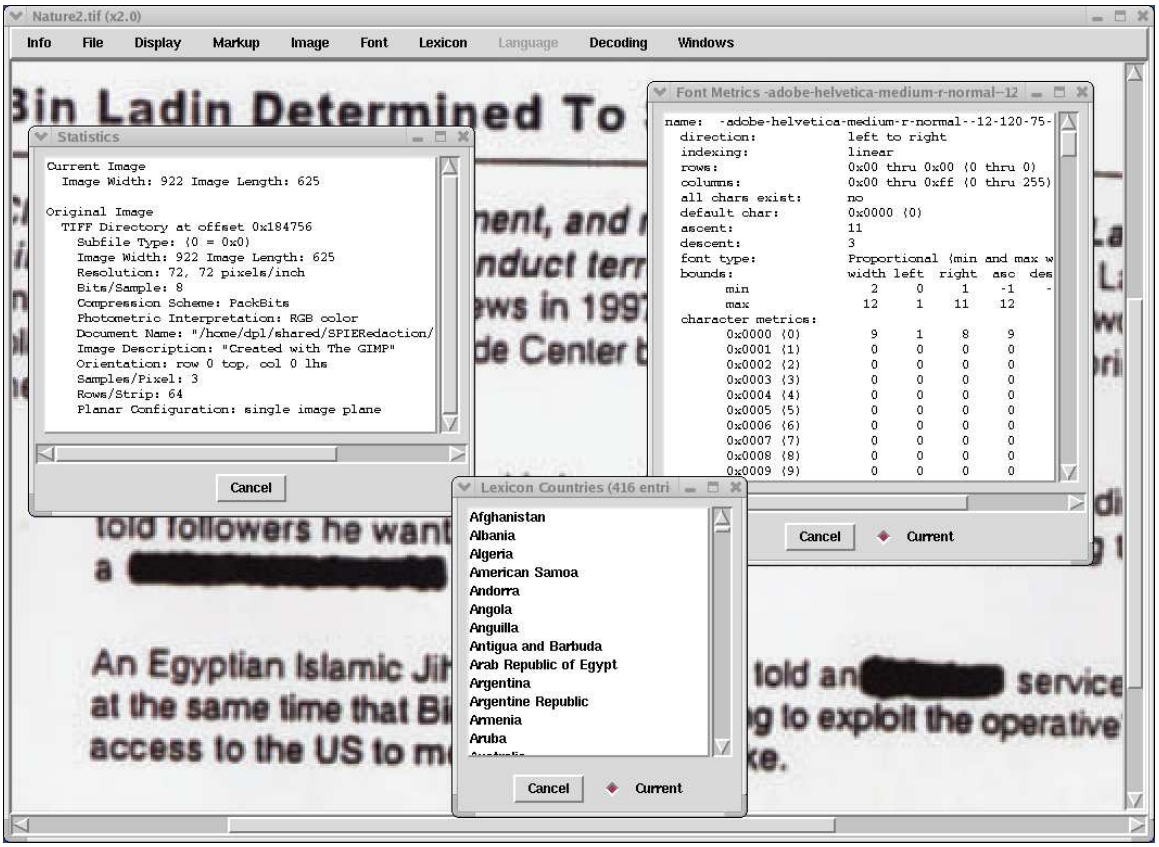

Figure 3: Screen snapshot of the Plumber prototype.

image cues that are available. In the figure shown, the string "Arlen Specter" matches well, whereas the names of the other three senators of a similar length when rendered in the same font (Thad Cochran, Craig Thomas, and Wayne Allard) do not.

The current version of Plumber also includes a powerful "grep-like" search function based on Spitz's character shape code (CSC) concept [14]. As illustrated in Figure 5, the rightmost redacted region reveals that the hidden text begins with either a capital letter or an ascender character (represented in CSC as 'A'), is probably followed by one x-height character (represented as 'x'), an 'i' (where the dot has not been completely obscured), and then some unknown number of x-height characters. In our example, a search for the pattern "Axix*" through a lexicon of U.S. state names returns only two hits: Arizona (the correct answer) and Maine. Any of the six categories of character shape codes defined by Spitz is available to the user.

Plumber is still very much a prototype at this point. However, we can begin to anticipate likely usage scenarios. A sample workflow might be:

1. Scan document in 8-bit grayscale.
2. Prepare histograms of background (areas without redaction) and regions surrounding and including redactions.
3. Subtract histograms yielding distribution of pixel values in redacted areas.
4. Generate binary image of redacted area based on resultant histogram.
5. Binarize grayscale image.
6. Determine location and dimensions of redacts.
7. Identify font.
8. Submit surrounding text to OCR.
9. Identify topicality of document in order to select topic-specific redaction lexicon.
10. Attempt to determine part-of-speech tags for redacted material by application of a probabilistic parser to adjacent text. If this adjacent text has very low probability, and document is consistently drawn from same grammar (or not from a grammar), then we would expect redacted text to have similar probabilities (normalized for sentence length, etc.), enabling labeling of documents as being of questionable grammaticality.
11. From redacted area dimensions, topic-specific lexicon and part-of-speech constraints, reduce the number of possible candidates for redacted text string.
12. From statistics about the degree of ambiguity remaining after such techniques are applied, develop a simple metric of the resistance of the redacted document to reconstruction.

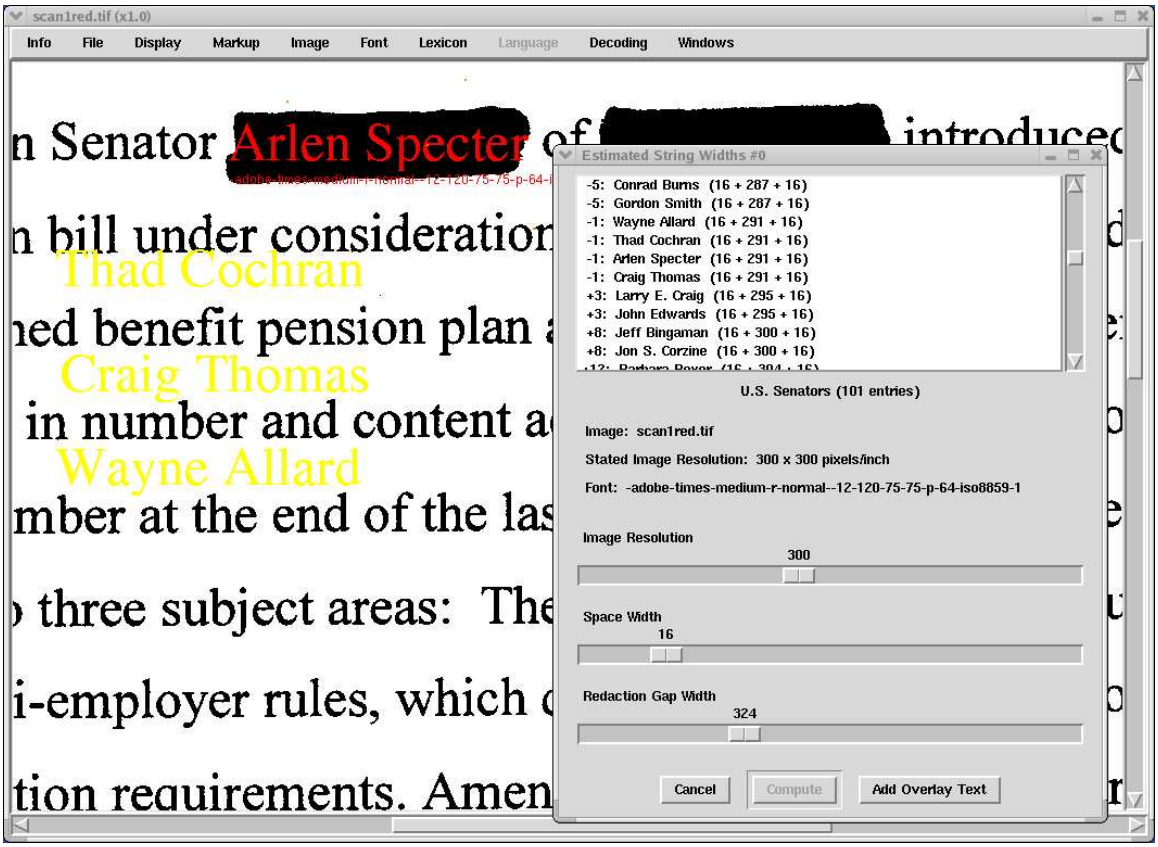

Figure 4: Overlaying potential redaction decodings in Plumber.

6. CONCLUSIONS

In this paper, we have examined ways in which sensitive information might leak through the process of redaction. Such attacks apply known methods from document image analysis and natural language processing. Once these weaknesses have been systematically identified and tested, effective countermeasures can be designed; our Plumber prototype is a first step in this direction.

7. REFERENCES

[1] Adobe Systems Incorporated, San Jose, CA. *Adobe Font Metrics File Format Specification*, October 1998.

[2] D. Butler. US intelligence exposed as student decodes Iraq memo. *Nature*, 429:116, May 2004.

[3] C. Fang and J. J. Hull. A word-level deciphering algorithm for degraded document recognition. In *Symposium on Document Analysis and Information Retrieval*, pages 191–202, 1995.

[4] J. J. Hull, S. Khoubyari, and T. K. Ho. Visual global context: Word image matching in a methodology for degraded text recognition. In *International Conference on Pattern Recognition*, pages 665–668, The Hague, Netherlands, 1992.

[5] S. Khoubyari and J. J. Hull. Font and function word identification in document recognition. *Computer Vision and Image Understanding*, 63(1):66–74, 1996.

[6] G. E. Kopec. Least-squares font metric estimation from images. *IEEE Transactions on Image Processing*, 2(4):510–519, October 1993.

[7] J. W. Leonard. Classified National Security Information Directive No. 1, September 2003. http://www.archives.gov/isoo/policy_documents/eo_12958_implementing_directive.html.

[8] D. Lopresti and A. L. Spitz. Information leakage through document redaction: Attacks and countermeasures, July 2004. Submitted for publication.

[9] D. Naccache, May 2004. Private communication.

[10] U.S. Census Bureau: Name Files, September 2004. http://www.census.gov/genealogy/names/.

[11] J. K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley, Reading, MA, 1994.

[12] U.S. Census Bureau: U.S. and World Population Clocks, September 2004. http://www.census.gov/main/www/popclock.html.

[13] A. L. Spitz. Using character shape codes for word spotting in document images. In D. Dori and A. Bruckstein, editors, *Shape, Structure and Pattern Recognition*, pages 382–389. World Scientific, Singapore, 1995.

[14] A. L. Spitz. Progress in document reconstruction. In *International Conference on Pattern Recognition*, pages 464–467, Quebec City, Canada, 2002.

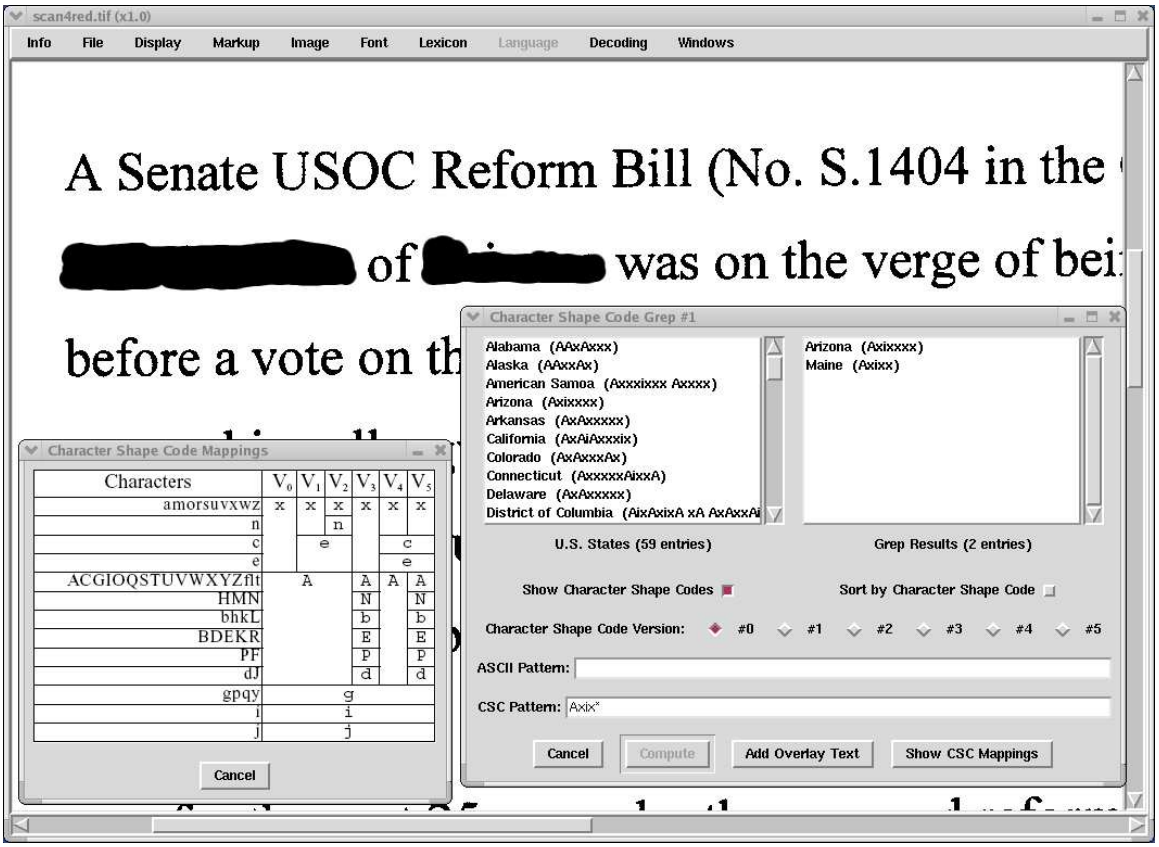

Figure 5: Character shape code pattern matching in Plumber.

[15] A. L. Spitz and J. P. Marks. Measuring the robustness of character shape coding. In *Proceedings of the IAPR Workshop on Document Analysis Systems*, pages 19–28, Nagano, Japan, 1998.

[16] Tcl Developer Xchange, September 2004. http://www.tcl.tk/.

[17] YAWL (yet another word list), September 2004. http://metalab.unc.edu/pub/Linux/libs/yawl-0.3.tar.gz.

[18] A. Zramdini and R. Ingold. Optical font recognition from projection profiles. *Electronic Publishing*, 6(3):249–260, 1993.