Cryptographically secure page numbering in LaTeX

William Gunther
Epic
wgunther@epic.com

Brian Kell
Google, Inc.
bkell@google.com

April X, 20XX

Abstract

Security in the modern world is of national importance. Personal information is being collected, stored, and analyzed by many governmental agencies, both foreign and domestic, as well as companies and organizations. People are looking to cryptographic solutions to keep their information safe. Yet the printed word has not changed in its security since the early 15th century. This paper makes the first steps toward bringing modern cryptography into the 1470s by introducing a LaTeX macro package to enable cryptographically secure page numbers. No longer will your document allow random access into your data. Our LaTeX package uses modern, secure hashing algorithms to prevent unauthorized access of page numbers by eavesdroppers and adversaries.

Contents

1 Introduction

In light of recent security breaches, secure communication has become more important than ever. Recent advances in cryptography have found such diverse

applications as online communication, commerce, information storage hardware, and voice correspondence. However, to the best of our knowledge, strong cryptographic techniques have, surprisingly, not yet been applied to page numbers.

Page numbers are ubiquitous and are quite important for page-based communication. In this paper we introduce a new LaTeX macro package, `secure-page-numbers.sty`, that enables cryptographically secure page numbers.

In addition to the immediate impact of protecting page numbers against unauthorized access, we hope this work will inspire future research on securing such things as section numbers, figure numbers, and citation numbers. As a provisional security measure until such work has been done, our macro package provides facilities for securely redacting this sensitive information.

The remainder of this paper is structured as follows. In Section X on page 0x425d22427978b6bc28a67815c2a61353e20a5c91861a656ac8ca6faa79299ec0, we describe the problem in more detail. Some notable previous work on the problem has been done; we give a brief overview of this work in Section X.X on page 0xc2082a19550b2a2c2b0b95cbdcb432a693ce0c72109e2d867cb3cb0e3d11ffe9. Our new method is described in Section X on page 0xc2082a19550b2a2c2b0b95cbdcb432a693ce0c72109e2d867cb3cb0e
Details about our implementation are given in Section X.X on page 0x654c937186a719b8cf18de2956b9b1ec81e418
Some brief initial observations about protecting other numbers in a document, such as section numbers, are described in Section X.X on page 0x654c937186a719b8cf18de2956b9b1ec81e4189173
and additional security recommendations appear in Section X.X on page 0xce65a4631ce9cd559d11180dcf7562f398
Finally, we conclude in Section X on page 0xce65a4631ce9cd559d11180dcf7562f398f61c69ca659896986ddaa0d7faf
and suggest some avenues for future work.

X Background

Page numbers have been a common feature of printed material since the 1470s. However, their nearly universal use is fraught with security issues that are often overlooked.

For example, if an eavesdropper merely glances over the shoulder of a person reading a sensitive document, the page number is instantly visible. If the eavesdropper can also get the last page number of the document, then she can quickly compute the percentage of the document that has been read. Two such observations, timed carefully with a stopwatch or large sundial, can provide the eavesdropper with enough information to estimate the remaining reading time. Armed with this knowledge, the eavesdropper can commence other attacks, knowing that the victim will be busy reading the sensitive document.

Many organizations and individuals use paper shredders to destroy sensitive documents. But, because of their small size, page numbers are often incompletely destroyed by these shredders. Unsecured page numbers leak information about the order of the scraps, which can aid a snooper who is trying to reconstruct a shredded document. The mere presence of a scrap bearing a large page number like 438 reveals to an adversary the potentially sensitive information that a lengthy document has been shredded.

If documents are printed on both sides of the page, these problems are

compounded: a page number reveals information about not only its own side of the page, but the other side too. In many documents, right-hand pages have odd page numbers while the page numbers on left-hand pages are even. So an eavesdropper in possession of an unsecured page number gains information not just about the order of the pages or the length of the document, but even the geometric position of the page.

X.X Previous work

One of the earliest techniques for page-number encryption, and by far the most widely employed today, is a Roman cryptosystem for page numbers in particularly sensitive early sections of a document, such as prefaces, forewords, and tables of contents. Unfortunately, while this algorithm is widespread, it has never been supported by strong cryptographic justification. In recent decades, concerted efforts by many researchers have identified significant security flaws in this method. For example, Hagenfried et al. [X] describe an attack that was able to recover the plaintext "24" from the encrypted page number "xxiv" in less than seven hours on commodity hardware.

Some of the security issues that arise with page numbers in two-sided documents were addressed indirectly in the work of Möbius [X], which was later extended by Klein [X]. However, one difficulty with these approaches is that the constructions they describe are non-orientable, which implies that they cannot be used for documents written in languages such as Chinese or Japanese. Our method is free from such regional restrictions.

X Secure page numbers in LaTeX

In order to address these issues, we propose a new method of securing page numbers in documents, using cryptographically secure hashing algorithms to replace all page numbers with salted hash values. In addition to obscuring the plaintext page numbers, this technique also produces much longer strings, which will be more likely to be obliterated by shredding.

Our LaTeX macro package is called `secure-page-numbers.sty`, available upon request from the authors. It can be included in any LaTeX document with the command `\usepackage{secure-page-numbers}` in the preamble, and secure page numbers are enabled with the command `\pagenumbering{shahash}`.

Once secure page numbers are enabled, all page numbers in the document will be replaced by cryptographically secure hash values. This provides a convenient drop-in solution providing maximum security.

The use of a one-way hash function means that it is possible for an authorized reader of the document to verify a hunch about what page she is on, by computing the hash of the hunch, but an eavesdropper cannot go backward from the hash to the page number.

Additionally, an eavesdropper cannot use the last page number to determine the length of the document. Gone are the days of reading *War and Peace* on

the bus and having a stranger say, "Wow, that's a big book!" With cryptographically secure page numbers, an eavesdropper can't tell whether it's a long novel or a short pamphlet.

X.X Implementation details

Our implementation uses the SHA-256 hash. It features a modular architecture, however, so that a different hashing algorithm can be substituted if desired.

In order to thwart rainbow-table attacks, a salt is appended to the page number before it is hashed. We are currently using the LaTeX job name as the salt. For example, the salt used for this paper was `sigbovik`.

Initially we attempted to implement the SHA-256 algorithm entirely in TeX macros. However, we soon ran into difficulties, including the arcane design of the LaTeX system and memory limitations built into the TeX engine itself. We were surprised to find that TeX does not have native support for bitwise operations on 32-bit unsigned integers. Since we can see no innocent reason to omit such functionality from a typesetting system, we suspect this must be an intentional deficiency added by the NSA in order to weaken cryptographic algorithms in TeX documents. We urge further investigation into this potential security backdoor.

To circumvent this problem, our LaTeX package calls an external Perl script to generate the appropriate SHA-256 hashes from the salted page numbers. This does require that LaTeX (or pdfLaTeX) be called with the `--shell-escape` option.

Our current design requires a fixed upper limit to the page numbers in the document. We have set this limit to 500, but it can be easily increased by editing the Perl script. If the page numbers in a document exceed this value, processing of the document will halt with the error message

```
! tl;dr.
```

The elimination of this fixed limit is left as an improvement for future work.

X.X Protecting other numbers

A natural next step after securing page numbers is to secure other sensitive numbers in a document, such as section numbers, figure numbers, and citation numbers. If these numbers are not carefully guarded, they could be used by an attacker to learn partial information about the order of pages.

Our LaTeX package does not yet support securing these numbers with cryptographic hashes, but as a provisional security measure we have included macros to redact these numbers from the document. For instance, the redaction of section numbers can be enabled with the command `\redactsectionnumbers`. Section numbers have been redacted in this paper to illustrate the technique.

X.X Security recommendations

When properly used, cryptographically secure page numbers make it impossible for an adversary to determine the order of the pages of a document without an expensive $\Omega(n!)$ time brute-force attack. However, in order to support this security, it is important for the document to be kept as loose pages (preferably one-sided). We strongly recommend against the use of binders, staples, paper clips, file folders, and other external devices that may inadvertently reveal information about the proper order of the pages. To maximize security, it is best to take all the pages of many documents together and throw them at random into a big sack. Another effective randomization algorithm is to toss the pages down a flight of stairs or off a suitably tall cliff.

The proper disposal of a sensitive document is also important. Some paper shredders, often called *cross-cut* shredders, cut a document in two directions, yielding many tiny bits of paper. On the other hand, other shredders, so-called *strip-cut* shredders, merely slice the document into strips. If a page containing one or more hashed page numbers is fed into a strip-cut shredder in a certain orientation, it is possible that each one of these hash values will end up on a single strip, so a snooper can easily pull out individual strips and read off the complete hash. Of course, the fact that the page number has been hashed means that the snooper will have difficulty gaining any useful information from this, but to maximize security when using a strip-cut shredder it is best to print every page number twice, once horizontally on the page and once vertically, to ensure that at least one of them will be obliterated by the shredding process. (This recommendation applies equally well to all sensitive information in the document, not just the page numbers.) Our LaTeX macro package does not yet support vertically printed page numbers, but we expect that this functionality will become available in the near future.

X Conclusions and future work

In this paper we have proposed a new technique for securing page numbers in LaTeX documents using cryptographic hashing algorithms. We introduced the macro package `secure-page-numbers.sty`, which implements this technique. We believe that this package has broad potential to significantly increase the security of page numbers in sensitive documents, and we urge all authors and publishers to adopt the use of secure page numbers as soon as possible.

One potentially fruitful extension to this work, which we have not yet explored, is the use of a public-key cryptosystem such as RSA for the encryption of page numbers in two-way communication. This is likely to be more useful than one-way hashes, since each party will be able to decrypt the page numbers in the document received from the other party.

In accordance with the first of our security recommendations above, it may be beneficial to scramble the pages of the document not only after they have been printed but also in the PDF itself. A LaTeX package to achieve this would

be difficult because of the design of the TEX engine, but we believe it may be possible with the aid of an external tool. As a workaround until this becomes available, we recommend storing large PDFs on a sequence of unlabeled floppy disks, which can then be shuffled together.

References

[1] Hagenfried, E. W., McPaulsen, F. J., and Svendt, P. Exploiting pseudofrequency eigenquotients to attack page number encryption. *IEEE Trans. Crypto. Splat.* 22(1), pp. 0x2fd5f03a787301a109748c1243f85f6522c3c862be9e1b156c84ef8dfaf9c296–0xbfbd44e30f1d9beef64f9324993b67e99bb1688ed77f474c46e58b2aca9c23dc, November 1999.

[2] Klein, F. *Über Riemann's Theorie der algebraischen Functionen und ihrer Integrale: Eine Ergänzung der gewöhnlichen Darstellungen*, pp. 0x2fa9b01782ee3680c2682f88fb03913ae03ff998602c90a0f8b2129567589535–0x9d7cf761c701b006aba29b8b4830eb46c78c28d8c4d448b4a5981a37697cb949, 1882.

[3] Möbius, A. F. *Werke*, vol. 2, pp. 0x3292d92ba775c2cac72f8e9091e04b3915d3ca7463c3c7b936ec7bede9933f9e–0x034c4ae2b61541e3a32124700b7c99c63974fb30d353649ec2fa828c5d90ec75, 1885.