Digital Whistleblowing in Restricted Environments

Graeme B. Bell

***Abstract*—The exposure of an organisation's illegal or unethical practices is often known as whistleblowing. It is currently a high-profile activity as a consequence of whistleblowing websites such as Wikileaks. However, modern digital fingerprinting technologies allow the identification of the human users associated with a particular copy of a leaked digital file. Fear of such discovery may discourage the public from exposing illegal or unethical practices. This paper therefore introduces the novel *whistleblower-defending problem*, a unique variant of the existing document-marking and traitor-tracing problems. It is addressed here by outlining practical steps that real-world whistleblowers can take to improve their safety, using only standard desktop OS features. ZIP compression is found to be useful for indirect file comparison, in cases where direct file comparison or use of checksums is impossible, inconvenient or easily traceable. The methods of this paper are experimentally evaluated and found to be effective.**

***Index Terms*—whistleblowers, whistleblower-defending problem, fingerprinting, document-marking, traitor-tracing, watermarking, wikileaks, social issues of digital information**

I. Introduction

THE SCENARIO: An individual discovers that illegal or unethical behaviour is taking place within their working environment. They have access to digital documents (email, PDF, DOC, JPEG . . .) that provide strong evidence of the undesirable activity. The individual wishes to expose the behaviour to an internal or external authority by 'leaking' a digital document, so that the behaviour can be remedied - i.e. *whistleblowing*. However, they fear that secret data uniquely identifying their own copy of the document - *a digital fingerprint* - may have been embedded invisibly inside the document (Wagner, 1983; Blakley et al., 1986). If present, a digital fingerprint could later be used to identify the whistleblowing individual, leading to reprisals against them. The dangers of detection can therefore strongly deter the whistleblower from serving their useful function in society. There is therefore a need for simple, appropriate, and inconspicuous techniques that allow digital whistleblowers to avoid detection, within the restrictions of monitored office environments.

Historically, research in the area of digital fingerprinting has addressed a different scenario, in which we are seeking to defend an innocent producer of media from an *opponent* (Wagner, 1983); a malicious and technically competent individual or group who freely share material anonymously. In research literature, studies have particularly examined the problems of *document-marking* (Brassil et al., 1994) and *traitor-tracing* (Chor et al., 1994). Those targeted by fingerprinting are assumed to be *traitors* or *pirates* - negative words, linked to unscrupulous activities. Research vocabulary also implies dark motives for those escaping capture: *collusion*, rather than *cooperation*; *maliciously attacking* fingerprinting systems rather than *countering* them. Even the term 'fingerprinting' derives from a process for catching criminals. In the cat-and-mouse game between researchers who construct or counter fingerprinting systems, there is an almost invariable assumption that those countering are playing the role of 'the bad guys'.

In this paper, a viewpoint is introduced that is unorthodox in two particular ways. Firstly, the malicious *traitor* is reconsidered as a socially-beneficial *whistleblower*. This realigns the moral compass of the problem, and yields a new perspective into this problem domain. It implies that there are circumstances where there is a genuine motive to find practical methods that help real-world whistleblowers to defend themselves from present and future document fingerprinting systems. In constrast, theoretical or proof-of-concept techniques may have limited utility for real-world whistleblowers. Secondly, this paper supposes restrictions upon the whistleblower's skills and actions, that cover an array of worst-case conditions that might exist in corporate or organisational environments. These restrictions significantly change the technical challenge of countering digital fingerprinting. For example:

- In many organisational I.T. environments, it is not possible to install or run arbitrary applications.
- The typical whistleblower is an ordinary member of society - they will almost certainly be naive about the operation of digital fingerprinting technology.
- The whistleblower is assumed to be working in a standard office operating system environment, such as a recent edition of Microsoft Windows or Mac OS.
- The whistleblower's actions and communication with colleagues may be under scrutiny.

This constrained scenario is henceforth referred to as the *whistleblower-defending problem* - a novel scenario that shares a connection with the previously studied document-marking and traitor-tracing problems. This paper describes and then addresses the whistleblower-defending problem.

This paper is structured as follows. Section II introduces whistleblowing, fingerprinting and the research context. Section III introduces this paper's assumptions. Section IV describes this paper's approach to the whistleblower-defending problem. Section V presents the proposed methods of this paper, and evaluates them experimentally. Section VI summarises results, presents conclusions, and is followed by references.

II. Whistleblowing and Digital Fingerprinting

A. What is Whistleblowing?

Whistleblowing is the practice whereby an individual reports inappropriate, unethical or illegal behaviour within an organisation, to an authority that is capable of correcting that behaviour. For example, if someone knew that their manager's actions were dangerous or fraudulent, they could contact the

Email: wbpaper@graemebell.net. Web: http://graemebell.net

company CEO or a government body and report it. The word 'whistleblowing' itself derives from the behaviour of British policeman, who in the past would blow a whistle loudly to alert other nearby policeman and members of the public to the fact that a crime was being committed.

Individual whistleblowers are not always motivated by virtue, but regardless, whistleblowing helps to correct serious problems in society. Fraud, human rights violations, health and safety issues, environmental damage and many other social wrongs are reported by whistleblowers every day - in fact, organisations such as the Association of Certified Fraud Examiners (ACFE) and the American Institute of Certified Public Accountants (AICPA) report that the primary means by which organisational fraud is prevented is through 'tip-offs' provided by whistleblowers (Ratley, 2008; AICPA, 2005). Modern societies rely so much upon the actions of whistleblowers, that 'whistleblower's rights' have become fully enshrined within national legal systems. In the U.S.A., examples include the Sarbanes-Oxley Act of 2002, Sections 301 & 806, and the Whistleblower Protection Act of 1989, 1994 & 2007 (U.S. G.P.O., 2002, 2007).

Most whistleblowers are simply ordinary members of the public, or regular employees within a company, who have become aware that immoral or illegal behaviour is taking place. They are seldom computer specialists or lawyers; rather, they are the everyman we might encounter on the street each day. Remarkably, a global survey conducted by Ernst & Young estimated that 20% of all workers in the USA had knowledge that would allow them to whistleblow workplace crimes (Ernst and Young, 2002).

Unfortunately for whistleblowers, whistleblowing is a word that is also strongly associated with *reprisals*; that is, acts of revenge carried out by the party at fault, once caught. Some individuals lose their job; others find themselves harassed and persecuted even after they have left the organisation. Examples of the historical 'rewards' for whistleblowing are found within the Sarbanes-Oxley Act, section 806:

> "Whistleblower protection for employees of publicly traded companies - No company ... may discharge, demote, suspend, threaten, harass, or in any other manner discriminate against an employee ... because of any lawful act done by the employee" (U.S. G.P.O., 2002)

Laws are not enough to prevent reprisals from taking place. Many whistleblowers are only prepared to come forward with information if their anonymity can be guaranteed. Ernst & Young's 2002 global survey on fraud found that 39% of public respondents were more likely to whistleblow on fraud if they could remain anonymous in doing so (Ernst and Young, 2002). Indeed, many countries now have laws requiring that whistleblowers must be provided with anonymous means to report crimes. Through the guarantee of anonymity, whistleblowers are encouraged to step forward instead of remaining silent. A good example of this is seen in Section 301 of the Sarbanes-Oxley Act:

> "Each audit committee shall establish procedures for the confidential, anonymous submission by employees ... regarding questionable accounting or auditing matters." (U.S. G.P.O., 2002)

It is clear that anonymity is very important in encouraging whistleblowing. However, whistleblowing itself is not always a single event, and there are many ways in which the cover of anonymity may be broken, besides the initial tip-off. Particularly, it may occur during followup communication, which is often an essential part of whistleblowing investigations (AICPA, 2005). After their initial report, a whistleblower may be asked to provide documents or other material to assist in the investigation. This is known as *leaking* documents. The whistleblower is at risk if a leaked document exposes their identity in any way.

In summary, then: societies strongly depend on whistleblowers to help prevent fraud, protect the environment, report crime, and generally assure that society operates in a legal and moral way. Yet whistleblowers seldom find any reward for their actions, and frequently face reprisals, despite laws that are meant to protect them. It is therefore important to find techniques that help to protect whistleblowers and encourage whistleblowing. The most reliable means by which we can do this is to ensure the anonymity of whistleblowers. Given that there has been more than twenty years of research into removing anonymity from users of digital information, this is far from a trivial task.

B. What is Fingerprinting?

Fingerprinting is the idea of placing or identifying *marks* on objects, so that they (and the people associated with them) can be uniquely identified. The idea of fingerprinting electronic data is said to date back to the early 1950's (Cox and Miller, 2001), but it was not until Wagner's landmark paper of 1983 (Wagner, 1983) that it became a popular field of study. Nowadays, *digital fingerprinting* refers to the modern technologies that mark digitally represented documents and data, so that they may be uniquely identified and associated with individuals. The word 'fingerprinting' itself originally comes from the fact that each human leaves unique oily marks on touched items. These marks facilitate the subsequent identification of criminals.

Fingerprinting technologies have been developed for copyrighted media, so that those who share their access to media illegally can be later identified. Generally, each copy of a fingerprinted document contains a number of marks, which taken together form a unique metaphorical fingerprint associated with those that are given access to that particular copy of the document. If copyrighted material is subsequently pirated, then one of the illegal copies is obtained by the copyright holder and the marks are analysed, yielding the unique fingerprint of the individuals responsible for leaking their copy of the media.

Key problems in this area include the *document marking problem* (Wagner, 1983; Blakley et al., 1986; Brassil et al., 1994; Maxemchuk, 1994) and the *traitor tracing problem* (Chor et al., 1994; Pfitzmann, 1996; Fiat and Tassa, 1999). Document marking is the problem of making changes to an electric document that will reliably allow the recipient of the document to be traced. The primary application of document marking is copyright control, but it has applications within any environment where access to data must be tracked; e.g. the military, government, and private research organisations. Traitor-tracing was originally a specific problem introduced by Chor et al, who studied the case of encrypted broadcast media, where unique decryption keys are provided to broadcast subscribers (Chor et al., 1994). Since then, the ubiquity of digital media has led to the idea of traitor-tracing extending far beyond the study of broadcast systems.

Research in these areas has operated under the assumption that there are two opposed, *technically literate* groups, each with effectively no limits on their time, skill or resources. The first group is attempting to mark documents to identify traitors. The second group are skilled *traitors* or *pirates* who are attempting to identify marks and render them useless. In academic studies, these roles are taken on by groups of researchers who engage in a friendly but competitive 'cat-and-mouse' approach to research.

C. Ad hoc fingerprinting approaches

In the real world, whistleblowers are also exposed by simpler means. Consider a media producer, whose material is being pirated and who has no special training in digital fingerprinting technologies. They may invent an ad hoc scheme to identify each copy of their work, by creating semantically-equivalent variants that are unique to each purchaser or user of the material. This might be done by manually varying the choice of fonts, words, characters, images or layout of the document until a sufficient variety of unique but equivalent documents are produced. The different versions of the document might then be manually recorded as being associated with a particular individual. Should one of the variants be observed in an un-permitted context, the associated user can be identified directly from the marks that were manually added. Some examples of such approaches are shown in Figure 1. By replacing synonyms, re-ordering phrases or altering layout, the naive or ad hoc fingerprinter can produce alternative, visually similar and semantically equivalent forms of the document. The method may vary to suit the format of the document; e.g. in spreadsheets, the producer may choose to vary some numbers slightly, to uniquely characterise each copy. In images, small visual perturbations may be made.

It is also possible that unique changes to a file may even be accidentally caused by the whistleblower themself while in possession of the document. Such changes might retrospectively be discovered and used as a digital fingerprint.

These techniques may seem trivial to specialists in signal processing, but would imaginably be quite effective in practice against non-specialists, who represent by far the bulk of potential whistleblowers. Ad hoc techniques of this nature were well documented in early research literature (Wagner, 1983), and an excellent overview of the remarkable inventiveness of ad hoc systems over the last 60 years can be found in (Cox and Miller, 2001). The advantages of such techniques are that they are trivial to implement, unpredictable for whistleblowers, and may be recovered even if the material is quoted in a non-digital medium. A disadvantage is that they may be noticed by an astute whistleblower, even without computer analysis.

"This is the newest book for old and young."

(a) Original unmodified example text.

"This is the latest book for old and young."

(b) Text with synonym replacement.

"This is the newest book for old and young."

(c) Font size variation, exaggerated.

Fig. 1. Examples showing ad hoc or informal fingerprinting.

In the early years of fingerprinting research, efforts were made to formalise ad hoc approaches and refine them; for example, techniques that hide fingerprint marks within aspects of the page layout that are imperceptible to human vision, yet obvious to machine analysis (Low and Maxemchuk, 1998; Brassil et al., 1994). Mathematical efforts were undertaken to increase the difficulty of fingerprint removal, for example by raising the number of uniquely fingerprinted documents necessary to discover all the marks (Blakley et al., 1986). Ultimately, academic research has produced systems of increasing sophistication that greatly reduce the risk of casual detection of fingerprints, and improve the chance of successful recovery of fingerprint data. The primary achievements of research have included:

- *subtlety* - particularly spread-spectrum watermarking (Tirkel et al., 1993; Cox et al., 1997) and steganography (Katzenbeisser and Fabien, 2000; Provos and Honeyman, 2003; Kessler, 2004).
- *resistance to damaging manipulations* e.g. robust approaches (Tirkel et al., 1993; Cox et al., 1997, 1996; Heintze, 1996; Nikolaidis and Pitas, 1998; Brassil et al., 1999; Park et al., 2001; Wu and Liu, 2002; Lee et al., 2006; Tang and Wang, 2008; Lin and Wu, 2008).
- *resistance to subversion by groups of 'traitors'* e.g. collusion-secure schemes (Boneh and Shaw, 1995; Low and Maxemchuk, 1996; Trappe et al., 2002, 2003; Wang et al., 2004; Wu et al., 2004; Lee et al., 2006; Lin and Wu, 2008).
- *asymmetry* - where fingerprint construction and tracing are kept distinct, addressing concerns about the legal validity of fingerprints, and about fingerprint reliability if the tracing scheme is successfully attacked (Pfitzmann and Schunter, 1996; Pfitzmann and Waidner, 1997; Eggers et al., 2000).
- *characterisation of theoretical limits* - (Anthapadmanabhan et al., 2008).

Such approaches may mask the existence of a fingerprint by spreading the fingerprint data subtly throughout entire images or pieces of text, or throughout the document as whole. Through the inclusion of redundant data or error correcting codes, they allow reconstruction of fingerprints even when the fingerprint has been detected and tampered with. Mathematical proofs guarantee the identification of fingerprints even when many fingerprints are combined together to obfuscate them. These techniques pose the challenge of digital fingerprints whose alteration or removal lies vastly beyond the capability of a technically naive user in a restricted environment; and even beyond the capabilities of highly competent users in unrestricted environments.

D. Traitor-tracing vs. Whistleblower-defending

Unfortunately, the traitor-tracing and document-marking technologies that are used to catch copyright pirates can equally be used by corrupt organisations to identify whistleblowers who are exposing the organisation's misdeeds. Digital fingerprints may be placed in documents, intended to remove the anonymity of whistleblowers if the documents are leaked. The organisation may benefit either by making it known that it has removed anonymity, which discourages whistleblowing; or by covertly gaining the ability to apply reprisals if whistleblowing takes place.

This paper now diverges from the conventions of pre-existing research in several ways. Firstly, the 'traitors' we are helping are not copyright pirates - here, it is assumed that they are ordinary members of the public who are beneficially correcting a fault in some organisation through their actions. This significantly affects how seriously we treat the role of protecting the 'mouse' in our cat-and-mouse research.

Secondly, this paper proposes that in the real world as it is faced by everyday whistleblowers, matters such as technology, technical skills and freedom of communication and action are overwhelmingly weighted to the advantage of the document marker. In contrast, the typical real-world whistleblower is rather isolated and vulnerable. They are probably not highly competent with computers; nor are they likely to have the benefit of unlimited time, skill and resources. They may have only minutes in which to leak a document, and they may only have a basic Windows environment available to them, with no opportunities to add extra software. They may even be operating under the continual observation of colleagues, security cameras and computer audit logs. This significantly reduces the range of actions available to the 'mouse' in escaping the 'cat', and amplifies the underlying technical challenge of the problem.

Thirdly, in the real world, most organisations do not take the time to mark every office document with carefully catalogued traitor-tracing fingerprints, in the way that a Hollywood studio might mark each copy of a film. Instead, we might expect traitor-tracing fingerprint technology to be applied only when a whistleblower is already suspected to exist, or for the most important internal documents. The whistleblower's problem is therefore essentially a matter of determining when they are at risk of being caught.

Fourthly, unlike a copyright pirate, a whistleblower probably does not seek to leak everything they have access to. Ideally, they will leak any relevant clear documents and avoid leaking any fingerprinted documents. The key problem for the whistleblower in their restricted situation is therefore not the *removal* of marks, but merely the *detection* of marks. This issue is extremely important, yet surprisingly the problem of *detecting* fingerprints has been traditionally considered trivial in fingerprinting research, and so it has been given very little research attention. This is because of the prevailing assumption that the 'mouse' is a highly competent computer user operating with freedom in their own environment. For example, Jong-Hyeon (2000) suggests:

> "Suppose a digital image is distributed with fingerprints. If a group of users who got it compare their copies, they can easily discover all the marks".

"Easily"? Hardly! Real whistleblowers face three problems:

- *Ignorance of the threat.* They are unaware of the idea of digital fingerprinting or the risk it poses for them. Real world whistleblowers are generally not experts in digital document technologies. If they are unaware of the threat, they will make no attempt to detect or counter it.
- *Ignorance of solutions.* If warned of the threat of digital fingerprinting, they lack the technical skill to do anything about it. The average member of the public simply *does not know* how to compare two digital documents to determine if they are perfectly identical or not.
- *Impracticality of existing solutions.* Even if somehow we could warn the world's whistleblowers of the threat, and train every potential whistleblower in the installation and use of suitable fingerprint detection software, we still fail; the average person is unlikely to be able to apply their new knowledge without being caught. They lack administrator rights to install software; they cannot hope to memorise or implement entire computer programs; it may be difficult to remove the document from the office environment; they may be watched by colleagues or cameras; they may have only seconds to act; actions taken on the computer may be logged; and any unusual office behaviour may be very conspicuous.

Furthermore: each cat-and-mouse phase of existing research has usually involved the countering of a specific known marking technology; whereas here, any defensive method that is employed should be sensitive to all possible present and future schemes for embedding unique fingerprints in digital documents, i.e. *universal* fingerprint detection. Similarly, the method should also be *blind* (not dependent on knowledge of existing fingerprinting systems) or it will fail to detect ad hoc fingerprinting approaches invented outside of the research community, or new technological systems.

The crux of the problem is now identified. As researchers, how can we provide simple, effective methods that will protect naive whistleblowers who are working in restricted environments, but which will operate universally and blindly against all potential fingerprinting schemes? This is the *whistleblower-defending problem*, and the focus of this paper is to present a practical solution to it.

III. Fingerprint detection

This paper now considers a *candidate document*, a digital document that a whistleblower wishes to leak, but which may contain a digital fingerprint that could later identify them. Any digital fingerprint which is present may be very subtle. If an organisation is attempting to trace which employee is leaking information, documents might be sent to each employee with a unique fingerprint as small as a single binary bit of changed information in the file, depending on the fingerprinting technique used. The bit representation of a whistleblower's document may therefore be identical to everyone else's (either no fingerprint, or everyone shares the same fingerprint), or the data may vary in one or more binary bit positions from other copies (i.e. possibly a fingerprint). The values or semantic meanings of one or more data bits in particular positions may be used to identify the whistleblower uniquely. This is true regardless of the system employed for marking digital media.

A. *Single document*

Let us consider the case where a whistleblower has access only to a single copy of the document. If they have no access to any other copy - nor any co-whistleblowers to cooperate with - then currently there is almost no means by which a technically naive user within a restricted computing and working environment could covertly and reliably establish the presence of a digital fingerprint. Although some steganalytic tools do exist that might detect evidence of a steganography-based fingerprint (Steganography Analysis and Research Center, 2011; Provos, 2004), it is unlikely that the user would have sufficient skill, opportunity and fortune to find, install and successfully use such a tool without risk of detection. The whistleblower's only real chance might lie in observing ad hoc fingerprinting - perhaps noticing an unusual choice of word, spelling, capitalisation or whitespace character within their document. If a document is to be leaked in this circumstance, the whistleblower's best hope might be to convert the document to the simplest possible form - ASCII plaintext with no graphics - and hope that they have removed some or all of any digital fingerprints present in the document.

The situation of a whistleblower with access to only a single copy of the document while working in a very restricted environment will not be further addressed within this paper, as it seems too challenging for the time being.

B. *Two or more documents*

A far more favourable situation exists if the whistleblower has access (temporarily or permanently) to another copy of the document which may be used for comparison. Alternatively, the whistleblower should seek to find a co-whistleblower with whom they can co-operate. It is this situation of having access to two documents that this paper is intended to address, within the previously described assumptions. Note that the issue of collusion attacks by technically competent adverseries operating in *unrestricted* environments has been broadly addressed by existing research, e.g. (Boneh and Shaw, 1995; Low and Maxemchuk, 1996; Trappe et al., 2002, 2003; Wang et al., 2004; Wu et al., 2004; Lee et al., 2006).

If there is *unrestricted* access to two candidate documents on a single computer, and *unrestricted* and unmonitored access to software, then fingerprint detection is trivial for any whistleblower trained in computer programming or use of a UNIX shell. The two documents can be opened and compared byte-by-byte for differences with a tool such as diff or cmp (MacKenzie et al., 2003). MD5 checksums could be generated and compared (Rivest, 1992; Wikipedia, 2011). If there is any observable difference whatsoever between the files (or file checksums) of two visually identical documents, then a digital fingerprint can be suspected by the whistleblower.

However, these forms of attack on digital fingerprints are generally not realistic under the assumed conditions of the whistleblowing environment - that is, a naive user limited to inconspicuous, plausibly deniable behaviour and limited communication, in a restricted and monitored computing environment with limited computing applications. Using tools such as MD5, cmp or diff to compare two user's copies of a file is not plausibly deniable, is not inconspicuous and is not straightforward for normal users . These tools can also leave trails of evidence behind them (for example, command line history records, or buffered text that has scrolled offscreen).

This paper now proposes alternative whistleblower defence techniques that can operate successfully under the assumed restricted conditions.

C. *Assumptions for these techniques*

It is assumed in this research that at least two copies of the document have become available; perhaps temporarily (placing restrictions on the time available to test the document, and thus the complexity of attack); perhaps at different times, or on separate machines (placing restrictions on the ability to perform a 'byte-for-byte' comparison); and perhaps to two cooperating whistleblowers independently (placing restrictions upon the amount of information that can be communicated). It is assumed that the whistleblower is being watched, and unusual actions - such as running an entirely new program - may be noticed.

What is therefore desired to suit these circumstances is a fast, simple, easy-to-remember attack that may be carried out even on separated copies of the document, with minimal information being communicated. Further, all of this should be possible while remaining within the previously described situational constraints.

IV. Principles and Approach

Two supposedly identical copies of a document are available. In order to have a unique digital fingerprint, the bit representation of the data for each document must be different, even if the semantic visual appearance of the data seems to be the same when loaded into an office application. The bit representations are necessarily nonidentical whether it is an ad hoc or formal method of digital fingerprinting that has been employed. There may also be a variation in filesize between the two documents, if the underlying data is different.

We can immediately observe that any two documents which appear to be the same, but have different file sizes, can be suspected of containing a fingerprint (or some other unique data which might be used for an equivalent purpose later). This deals with the case where an ad hoc or formal fingerprinting approach carelessly produces fingerprinted documents with different sizes. While this observation is trivial to make, it is nonetheless remarkably helpful for real-world whistleblowers.

The next problem is more challenging: identifying the possible use of digital fingerprints in cases where the sizes of two files are the same, but where the internal data varies in some small way - but without having a tool to directly compare each bit of the data, or a checksum generator. This is the main challenge that this paper attempts to overcome.

Fortunately, a tool can be created using the file compression facilities available within the GUI of modern operating systems. In developing the methods of this paper, a straightforward theoretical concept will now be employed, namely that *'identical bitpatterns compress identically'*, whereas *'different bitpatterns compress differently'*. These principles are true for any deterministic scheme for compression. Here, it will be shown that file compression of the candidate documents provides a rapid and convenient estimate of whether two files are identical or non-identical.

It is quite undesirable to reproduce the problem of determining file equivalence after we have compressed the candidate documents, so we cannot expect the compressed files to be compared bit-by-bit. Instead, only a single metric will be compared - that is, the easily observed file size of the compressed forms of the two documents. Methods based on this approach will be seen to yield several desirable properties.

Sufficiency: It will be shown that the tests that are given in this paper, based on compression, are sufficient to determine if a fingerprint may exist with a high degree of confidence.

Availability: Compression facilities based on the 'ZIP' compression system are already built into the context menus of Microsoft Windows (XP, Vista) and Mac OS X. It would require a special and unusual effort for organisational IT teams to remove this part of these operating systems. It can therefore be taken for granted that these compression utilities will be available even in a very restricted computing environment.

Plausible deniability: Firstly, it is very plausible that an employee might seek to compress or archive some data, perhaps in order to email it, fit it onto some other storage, or in order to group a number of files in a more organised manner, as part of their normal everyday work. Secondly, notice that no new tools need to be installed to check for digital fingerprints - assisting the whistleblower in plausibly denying that they were ever considering leaking a document. Thirdly, at no point will it be possible to know the whistleblower was secretly carrying out file comparison, rather than file compression or archiving.

Simplicity/Ease of use: Firstly, even non-expert users are very likely to be already familiar with the idea of a 'ZIP file', and even the most naive document users are likely to be familiar with the idea of right-clicking on a file to perform operations. Secondly, the number of steps that are needed for tests is very small, making it easy even for inexperienced computer users to learn and memorise the technique. The tests are also very quick to carry out, which assists whistleblowers with limited access to the documents.

Independent testing: It will be seen that it is not necessary to have both documents on the same computer or at the same time in order to carry out the test.

Limited communication: It will be shown that by using compression as a form of ad hoc checksum, only a minimal and convenient amount of information needs to be remembered and shared between two cooperating whistleblowers, if the two documents are only available separately.

A. Theory: Compressed file size as an ad hoc checksum?

In order to use the compressed file size of files as a meaningful ad hoc checksum, it is necessary to investigate if a subtle variation between the data in two files reliably results in a user-noticeable variation in their compressed file sizes. It cannot be taken for granted that the compressed form of two different but equally-sized bitpatterns, under a fixed scheme of compression, will vary noticeably in size in a manner that indicates the presence of possible fingerprints. Consider that operating systems usually report file sizes to the user in units of whole bytes, yet fingerprinting may take place at a scale involving changes to individual bits. These scales are almost a full order of magnitude apart. We might even expect that two files which are almost perfectly identical, will usually compress to identical file-sizes (measured in bytes).

Essentially, we must ask: *If the values of one or more data bits are different between two nearly-identical data files, will compressing the files reliably result in an easily-observable variation in their size, measured in bytes?*

The minimal difference between two documents with unique digital fingerprints (or with/without a digital fingerprint) is 1 data bit. In practice, it is reasonable to expect the number of non-identical bits resulting from the embedding of digital fingerprints to be at least one or two orders of magnitude greater than this, perhaps 10 to 100 data bits, depending on the fingerprinting method in use. The first experiment of this paper seeks to establish the effects of such variations upon the compressed file size.

B. Experiment 1: Similarity of compressed file size

1) Purpose: This experiment investigates if it is likely that two slightly dissimilar files of equal size, will compress to an identical size (in bytes) after ZIP compression. If it is unlikely, then file compression may be used to detect fingerprints.

2) Method: Extracts from 10 PDF files relating to digital fingerprinting have been randomly selected from citeseer.ist.psu.edu. Each extract is approximately 1MB in size, (± 0.1MB). The extracts contain a mixture of digital media content - text, vectors, tables, and bitmaps, as well as varying degrees of use of internal data compression.

For each of the 10 test files, variant files were generated by pseudorandomly selecting n bit-positions in the file and varying the value. The smallest possible variation is a single 1-bit change within the 8 million bits available; the largest variation tested here is 256 altered bits within 8 million bits. This is believed to represent a challenging problem for ad hoc file comparison techniques. For each value of n, 100 different variant files were generated, compressed with ZIP, and compared to the ZIP'd size of the original file. In total, then, for each value of n, 1000 permutations were considered: 10 different PDF files, each with 100 different variations. This allows an estimation of the likelihood that an n-bit variation between two files will introduce an easily-observed change in the ZIP'd file size, measured in bytes. The values of n

Deflation	File	$n = 0$	$n = 1$	$n = 2$	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$	$n = 256$
84%	A.pdf	0%	92%	99%	100%	100%	100%	100%	100%	100%	100%
82%	B.pdf	0%	87%	97%	100%	100%	100%	100%	100%	100%	100%
70%	C.pdf	0%	79%	93%	98%	100%	100%	100%	100%	100%	100%
67%	D.pdf	0%	64%	87%	97%	100%	100%	100%	100%	100%	100%
16%	E.pdf	0%	19%	33%	59%	86%	94%	99%	100%	100%	100%
15%	F.pdf	0%	20%	30%	56%	86%	97%	100%	100%	100%	100%
10%	G.pdf	0%	12%	20%	36%	66%	87%	96%	100%	100%	100%
8%	H.pdf	0%	10%	24%	42%	69%	90%	97%	99%	100%	100%
5%	I.pdf	0%	3%	7%	18%	42%	68%	84%	100%	100%	100%
2%	J.pdf	0%	0%	3%	10%	24%	42%	69%	90%	98%	100%
36%	Mean	0%	39%	49%	62%	77%	88%	95%	99%	100%	100%

TABLE I

% OF ZIP-DETECTED VARIATIONS, WHEN n BITS OF A 1MB PDF VARY BETWEEN TWO FILES.

chosen are: 0, 1, 2, 4, 8, 16, 32, 64, 128, 256. The particular choice of these values is not significant, except for the case $n = 0$, which was tested to verify that the procedure was not producing false positive results in cases where there were no variations between files.

In total, 10 real-world files were tested, with 10 values of n and 100 different variant files, yielding 10,000 unique experiments.

3) Results: Table I shows the results of the experiment.

A *deflation* measure is provided indicating the degree of file size reduction achieved by the compression algorithm. A high deflation measure indicates the original material was primarily uncompressed data; a low measure indicates the original material contained primarily already-compressed or random data, which is difficult to compress further. Results are ordered from ‘poorly compressible’ to ‘easily compressible’ data. Notice that the experimental results measure the percentage of cases where dissimilarity was detected through a change in the observable file size (measured in bytes). 0% indicates that all variant files had the same compressed file size as the original file, measured in bytes; 100% indicates that no variant files had the same compressed file size.

C. Discussion of Results

Several observations can immediately be made from inspection of Table I.

- The average case result shows that by simply zipping two files and comparing the size in bytes, differences of more than 2 bits (in total) among 8 million bits are very likely to be noticeable. There is even an reasonable chance of detecting even a single 1-bit difference between a pair of 8-million-bit files, in the average case. This is a remarkable level of sensitivity, given that it is being achieved by a compression algorithm that was never intended for use as an ad hoc checksumming approach.
- The best case results are seen in the top rows of the table. These are files containing mostly non-compressed data e.g. plain text, uncompressed bitmap images. In such cases, even a single bit variation between two pieces of data is extremely likely to be noticed if you compare the file size in bytes of the corresponding ZIP files. An 8-bit change is essentially guaranteed to be observable.
- The worst case results are seen towards the bottom of the table. These are files containing already highly compressed data. However, even in the worst case, it is possible to determine with near complete certainty whether the data differs from the original, when more than 32 bit-positions have varying values, within 8 million bits.
- The results show that as more data bits vary between two files of equal size and with identical names, it very rapidly becomes unlikely that the ZIP-compressed size of the two files, measured in bytes, will remain the same.
- These experimental results suggest that there is a lower limit upon the number of bits that vary between two files, before 100% of variations in file data (of that magnitude) will result in observable changes in file size after compression with the ZIP algorithm, regardless of the data content. This limit appears to be at, or near, 256 changed bits among 8 million bits of data.
- The technique is most effective when the files being compared consist of mostly uncompressed data. It seems that it is easier to use this technique to detect variations in uncompressed data, than in highly compressed or random data.
- No false positive results were found.

D. Experiment 2: Different file types and file sizes

The files tested so far have been 1MB PDF files. It is possible that these findings are applicable only to PDF files, or only to files of approximately 1MB in size. In principle, the testing method is ignorant of the semantic meaning of the data being tested, so there is no reason to suspect that the file type should make a difference. However, we might either expect that the original size of the file will have no effect; or that as the ratio between the ‘amount of variation’ and ‘size of the file’ increases, it will become easier to detect variation. It is best to explore these issues experimentally.

1) Purpose: This experiment investigates if the results found in Experiment 1 can be expected from other types of media besides PDF files. The most common real-world office documents, besides PDF, include plain text (email/web pages), Microsoft Excel files (.xls), Microsoft Word documents (.doc), and Microsoft Powerpoint presentations (.ppt).

Defl.	Size	Filetype	$n = 0$	$n = 1$	$n = 2$	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$	$n = 256$
79%	45 KB	.XLS	0%	93%	99%	100%	100%	100%	100%	100%	100%	100%
76%	34 KB	.HTML	0%	95%	100%	100%	100%	100%	100%	100%	100%	100%
50%	354 KB	.DOC	0%	57%	78%	93%	98%	100%	100%	100%	100%	100%
44%	1323 KB	.PPT	0%	52%	61%	68%	74%	83%	93%	98%	100%	100%
36%	1000 KB	.PDF	0%	39%	49%	62%	77%	88%	95%	99%	100%	100%

TABLE II
% OF ZIP-DETECTED VARIATIONS FOR RANDOMLY SELECTED REAL-WORLD MEDIA FILES OF VARYING TYPES AND SIZES.

This experiment also indirectly considers whether file size is an important factor in these results - i.e. are the results so far dependent upon the 1MB file size, or are they dependent upon the ratio between the number of varying bits, and the size of the files? These are important issues, as many real world documents may be only a few kilobytes in size.

2) Method: A search was carried out for the phrase 'digital fingerprinting' using the Google search engine, searching in turn for the file types '.doc, .html, .xls, .ppt'. Five randomly selected results on the first page of each search were downloaded and used as the test media for this experiment. As in the previous experiment, variant files were generated for each of the 20 test files that were gathered. 20 real-world files were tested, with 10 values of n and 100 different variations, yielding 20,000 unique experiments.

3) Results: Table II shows the results of the experiment.

When looking at the results, the most important question that should be asked is: do these results seem similar or dissimilar to what has been shown already, given different file types, and different file sizes? Corresponding results from the previous PDF experiment are included to allow comparison. For each file type, the mean values of size, deflation, and detection rate were calculated and are shown in the table.

E. Discussion of Results

The results were in line with the earlier experiments on PDF files, in terms of detection rate relative to the deflation figure for each file. The .xls and .html documents naturally tended to be uncompressed; the .doc and .ppt files used here each contained some degree of poorly compressible data.

Although the mean values shown may seem to suggest that a smaller file size might make it slightly easier to detect variations, in individual experiments it was noted that such an effect does not seem to exist. Only the compression level (deflation) was useful for predicting the ability to find variations between two files. The results appear to be unaffected by file size or file type; except for the fact that a given file type may imply a particular level of use of compression internally.

This experimental result can be conveniently verified on a home computer, as follows. Create a text file using Notepad (Windows) or TextEdit (MacOS), and copy a few hundred words of text into it[1]. Save the file as a '.txt' file. Next, zip-compress it, and note the size of the zip file. Delete the zip file. Then, change three or four characters within the document, without changing the total number of characters. This will typically alter approximately 2-3 binary bits of data per character changed, if you are altering letters of the roman alphabet. Save the file and zip-compress it again. Note the size of the new zip file. You should find that the two zip file sizes are frequently different, even though the original documents were identical in size and almost exactly identical in content other than a few dissimilar binary bits.

F. Experimental conclusions

These results impose constraints upon future work within digital fingerprinting and traitor-tracing[2]. If more than 16 data bits vary between two differently fingerprinted documents, in almost any size or type of file, then it is likely that the existence of the fingerprint will be detectable by the use of file compression (and thus detectable even for a naive user in a restricted and monitored environment). Even when only a single bit varies, in the majority of cases the fingerprint will be easily detectable via compression.

G. Theoretical behaviour of ZIP compression

The experimental results in this paper raise a number of puzzling questions. Firstly, why does ZIP reliably exhibit this sensitive behaviour when it is presented with two files that vary by only a tiny degree, perhaps having only a single bit value that is different between two huge files? Secondly, why do already compressed files exhibit less sensitivity to variations between them than uncompressed files? Finally, why do large files behave almost exactly the same as small files?

The ZIP program implements the DEFLATE algorithm (Deutch, 1996) for compression and decompression. DEFLATE has two parts: LZ77 compression (Ziv and Lempel, 1977) and Huffman coding (Huffman, 1952). Huffman coding looks at the statistics of a large piece of data, and uses this information to build a coding tree as a means of achieving compression (we can assign short paths within the tree corresponding to the most commonly used strings). To use an everyday analogy: if people frequently call the telephone operator, we should assign that person a short phone number i.e. 100 (see (Feldspar, 1997)). In contrast, LZ77 uses run length encoding, with a sliding window. It iterates through data, looking for byte patterns that have occurred recently in the past 32KB of data, that are occurring again at the current

[1] When verifying this result, it is important to use e.g. Notepad, Wordpad, or TextEdit with a simple text format such as '.rtf' or '.txt'. If a tool such as Microsoft Word is used instead, with '.doc' format, the size of the file may vary even when no characters are changed. This effect can make it difficult to verify that it is the data content - and not the file size - that produces the result.

[2] "Whistleblower-attacking"!

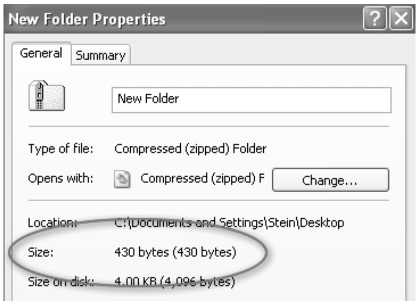
position. If a pattern of data is repeated, LZ77 refers back to the position of the original pattern. This saves space compared with rewriting the pattern of data again in full. To use an everyday analogy: instead of writing out Thursday's shopping list in full, we can simply write "buy the first 10 items on Tuesday's list again".

Firstly, can a single bit variation directly and significantly alter the behaviour of LZ77 as it iterates over the entire data? Not really; LZ77 sees only a 32KB sliding window as it looks for opportunities to save space. It is possible however, that a single bit variation between two files could have a significant *indirect* effect upon their corresponding output from LZ77, for two reasons. The first reason is 'amplification': LZ77 is a byte-stream compressor, not a bit-stream compressor. A change to a data bit therefore necessarily has consequences that are an order of magnitude larger. The second reason is 'knock on effects'. Not only is the LZ77 output affected while encoding the byte with the varied bit (which may now refer back to some different position in the previous 32KB), but additionally, the encoding of any future bytes in the next 32KB which might refer back to the varied bit's position are affected; and then in turn, any future bytes that refer back to those neighbouring bytes; and so on. Essentially a chaotic 'butterfly effect' of small changes may take place. This may result in several small variations throughout the LZ77 output: particularly within the 32KB following the bit variation, especially around the bit position itself, and very occasionally, outside of the next 32KB as well. Each variation may add or subtract a few bytes from the total size of the output. Notice that such behaviour will result from *every* bit variation between the two files, leading to greater numbers of knock-on effects as more bits vary.

In the case of previously compressed or random data, there will be few recurring patterns that LZ77 can exploit, and consequently, few opportunities for a changed bit to affect the encoder's output except at a single byte position. Therefore, the total file size in this situation will rarely vary between the two files being tested. Regarding the matter of file size, we can consider that LZ77 operates within a 32KB window, rather than over the file as a whole. Consequently, the experimental results were not dependent on a large or small file size as LZ77 does not 'see' the whole file at once.

Turning now to Huffman coding, recall that Huffman trees represent the most commonly used phrases as the shortest codes. Is it likely that a single bit will significantly alter the overall statistical frequency of the most common phrases? Certainly not - the shape of the tree will be unaffected at the main branches. However, it is possible that as a result of the variation in the data, some uncommon string will move position within the coding tree (or perhaps enter or exit the tree), if we compare the encodings of files A and B. This small variation in the leaves of the coding tree may affect the data representation of the tree, as well as the output of the Huffman coding itself. These effects may amount to a variation in the ZIP output of several bytes in size, per 1-bit variation in the original file.

In the case of previously compressed or random data, the coding tree will already contain every possible code at some point in the coding tree, with no code left unused. All of the possible leaves of the coding tree will be present and will have modest non-zero probabilities attached to them. These are not easily affected by a single bit variation between the two files. Essentially, Huffman trees are more resilient and likely to retain their shape in the face of tiny perturbations, when they have been built from effectively random data. Regarding the issue of file size, we see that DEFLATE compressors split data into blocks and use separate Huffman coding trees in each block. This may again explain why the size of the files being tested did not seem to effect the results of the experiments in this paper.



New Folder

Type of file: Compressed (zipped) Folder

Opens with: Compressed (zipped) F

Location: C:\Documents and Settings\Stein\Desktop

Size: 430 bytes (430 bytes)

Size on disk: 4.00 KB (4,096 bytes)

Fig. 2. The size field that should be noted in Methods 1, 2 and 3.

H. Summary

For the purpose of this paper, it is sufficient to notice that generally, the ZIP-compression of two files of any kind with any variations in their internal data - even a single flipped bit - will tend to result in compressed files with a difference in their observed file size (in bytes). With this knowledge in hand, further methods can be created to support safer digital whistleblowing.

V. Whistleblowing: Methods

In this section, three methods are presented which may be used by co-operating whistleblowers in a restricted environment. Each of these methods has a reasonable chance of successfully detecting digital fingerprints. It is important that in each case, the user remembers to carry out the steps exactly as described: particularly, ensuring that the files are named identically.

A. Method 1: Compare file sizes

Method 1 is simple, but can sometimes detect formal and ad hoc fingerprinting techniques. For each document:

1) Right click on the file, and select 'Properties' (Windows) or 'Get Info' (Mac OS) from the context menu.
2) Make a mental note of the number in the 'Size' field, *measured in bytes*. If the files are available to separate people, or exist on separate machines, or are available at separate times, this number (the size in bytes) must be remembered and communicated (see Fig. 2).
3) Compare the file sizes of the candidate documents. If they are in any way different, between two copies of a file which are meant to be 'identical', then some variation is present in the files that might be later used to identify the whistleblower(s).

B. Why does this work?

Firstly, ad hoc fingerprinting may be clumsy, resulting in unequal amounts of data in the two documents (perhaps using synonym words of unequal length; whitespace representations of unequal data representation size; editing carelessly and so on). Secondly, certain file types such as office documents may involve some degree of compression as part of the file type standard. In these cases, the subtle variations representing fingerprints may produce the 'unequal file size' property discovered in Experiment 1, directly because of the file type's internal use of some compression algorithm.

C. Attacks on Method 1

Developers of digital fingerprinting systems might reasonably anticipate an attack of the form described in Method 1. They may have rate-adaptive compression systems, or padding systems, that allow files such as JPEGs or PDFs to have a fixed size, regardless of any embedded fingerprint; or they may generate multiple potential versions of fingerprinted documents and exclude any that could be exposed by Method 1. Nonetheless, Method 1 is easy to remember, takes only a moment and can detect some ad hoc and formal fingerprinting methods. It can indicate that a fingerprint may be present. It *cannot* prove that a fingerprint is not present.

D. Method 2: Compare zipped file sizes

Method 2 is slightly more complicated but much more effective. It uses file compression directly, to allow file comparison.

1) For each candidate document:
   a) Make sure the files to be tested have exactly the same innocuous filename, i.e. "PrintMe.doc". This may involve placing the documents into different folders, if they exist on a single machine.
   b) Right click on the candidate file's icon, and select 'Add to zip', 'Compress files', or 'Send To/Compressed (zipped) Folder', depending on the OS version. A zip file will be generated in the same location as the file being tested. In Windows XP, it may appear as an unusual folder icon.
   c) Note the size of the resulting zip file/folder in bytes by right clicking on the new file and selecting 'Properties' or 'Get Info'.
2) Remember or communicate the file size, if necessary. Compare the file sizes of the zip files. Again, two files which are meant to be 'identical' should not produce different results; it may suggest that changes are present which could later identify the whistleblower(s).
3) Delete the zip files that were generated during the tests and empty the recycle bin. Return the files to their original filenames and locations.

SCCD : <u>S</u>ame Name, <u>C</u>ompress, <u>C</u>ompare, <u>D</u>elete.

E. Why does this work?

The experiments earlier showed that if we have two non-identical data representations of a file, then after compression, the resulting zipped file is quite likely to be different in size. Method 2 exploits this phenomenon directly to try to determine if a digital fingerprint is present. Method 2 is quick to carry out, and may detect both ad hoc and formal fingerprinting methods. It can quickly indicate that a fingerprint may be present. It *cannot* prove that a fingerprint is not present.

F. Attacks on Method 2

Developers of future fingerprinting systems might attempt Method 2 automatically on a number of potential fingerprinted documents, and rule out those fingerprints that cause the resulting zipped documents to expose the fingerprint. In essence, the system would keep only those 'lucky' fingerprints that cause Method 2 to fail. Therefore, although it is useful now, Method 2 might be less useful in future against improved fingerprinting systems. However, it is reasonable to expect Method 2 will always remain effective against ad hoc fingerprinting techniques, as well as some formal techniques.

G. Method 3: Prepended text and zipped file size comparison

Method 3 is the most complicated method presented here. The result of Method 3 cannot be predicted a priori by the designers of fingerprinting systems, as it uses random input from the whistleblower. This property makes the technique more robust against future fingerprinting systems. However, this method involves actions that may be perceived as unusual in some workplaces, which reduces plausible deniability and may draw attention to the whistleblower.

1) For each candidate document:
   a) Make a copy of the document. Either select the document, then press CRTL-C, then CRTL-V (Windows), or select "Duplicate" (Mac OS).
   b) Open the copy in "Notepad" (Windows) or "TextEdit" (Mac OS). If the file is large, you may need to use "Wordpad" (Windows)[3].You will need to use 'Open with' from the right-click menu, rather than 'Open'. Document types such as .xls, .pdf and .doc may cause junk text characters on the screen. You may safely ignore the junk text.
   c) Type a random short phrase into the start of each file - something that does not identify you and is inconspicuous, e.g. "quit". Save the file. Use the same short phrase for each file.
   d) Make sure each copy has the same name (as before). Zip each copy of the files, as in Method 2, from the right-click context menu.
   e) Note the file size in bytes as before, using 'Properties' or 'Get Info' (right-click, context menu).
   f) Delete the zip files that were generated during the test, and also the extra copies of the documents that you made. Empty the recycle bin.

[3] It is important that editing is carried out with these simple text editing tools rather than complex tools such as Microsoft Word or Microsoft Excel. Complex tools may alter the size of the files in an unpredictable way even when identical actions are taken; this unpredictability will make the technique worthless by producing many false positive results.

Deflation	File	$n = 0$	$n = 1$	$n = 2$	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$	$n = 256$
84%	A.pdf	0%	96%	100%	100%	100%	100%	100%	100%	100%	100%
82%	B.pdf	0%	98%	99%	100%	100%	100%	100%	100%	100%	100%
70%	C.pdf	0%	88%	97%	100%	100%	100%	100%	100%	100%	100%
67%	D.pdf	0%	79%	96%	100%	100%	100%	100%	100%	100%	100%
16%	E.pdf	0%	26%	38%	68%	90%	99%	100%	100%	100%	100%
15%	F.pdf	0%	24%	35%	67%	92%	99%	100%	100%	100%	100%
10%	G.pdf	0%	16%	27%	53%	79%	93%	98%	100%	100%	100%
8%	H.pdf	0%	15%	37%	61%	79%	96%	99%	100%	100%	100%
5%	I.pdf	0%	4%	9%	21%	47%	75%	92%	100%	100%	100%
2%	J.pdf	0%	10%	16%	35%	62%	82%	96%	98%	100%	100%
36%	Mean (Method 3)	0%	46%	55%	71%	85%	94%	99%	100%	100%	100%
36%	Mean (Method 2)	0%	39%	49%	62%	77%	88%	95%	99%	100%	100%

TABLE III
% OF VARIATIONS DETECTED BY METHOD 3, WHEN n BITS OF A 1MB PDF VARY BETWEEN TWO FILES.

2) Compare the file sizes of the zip files you made. If they are different between two copies of a file which still ought to be ‘identical’ following an identical change being made, then a variation is present between the files’ data that might be used to identify the whistleblower(s).
3) Finally, go back to the start and repeat the test a few more times, using different test phrases instead of ‘quit’. If any of the tests result in the pair of zip files having unequal file sizes, double check that the phrase was entered in an identical manner into each file. If the difference in file size persists after double-checking, there is some variation between the files and they should be suspected.

CSA-CCD : Copy files, Same name, Add word, Compress, Compare, Delete.

H. Why does this work?

If two files are identical, then the bitpatterns after prepending a randomly selected short phrase must also be identical and will compress identically. On the other hand, two non-identical files, with identical extra data prepended, will not compress identically and the compressed file size may vary unpredictably between the two files. Consider that a digital fingerprinting system has no way of anticipating which phrase might be added to the original document. It is therefore impossible for a digital fingerprinting system to ‘pre-test’ the zipfiles that might result from Method 3, to guarantee that fingerprints can never be detected. The following experiment investigates whether Method 3 is beneficial in practice.

I. Concerns with Method 3

The use of tools such as Wordpad, Notepad or TextEdit in the whistleblowing environment introduces concerns which merit further discussion. The first issue is conspicuity. While these tools may be normally used in some environments, they may not be normal in all environments. Furthermore, extra steps are involved in this technique as a consequence of repeatedly copying and editing the files. Whistleblowers should therefore consider in advance whether it is likely that their actions will be noticed if they use Method 3.

The second issue is plausible deniability. The whistleblower must be able to provide a reasonable explanation for their actions if noticed. With Method 2, this matter was relatively trivial, as file compression and archival are normal office tasks. With Method 3, the whistleblower may be asked why they are using a slightly unusual program, and why they are editing files. Possible explanations might include: ignorance that the wrong program is being used; an accidental misclick while opening the file; the advice of a third party to use the program; and ‘accidentally’ typing quit/exit while trying to get out of the program. Whistleblowers should therefore consider the matter of plausible deniability before adopting Method 3. Some whistleblowers may also wish to consider ‘misconfiguring’ an office system in advance so that the default action for opening document files is to use the ‘wrong’ program.

J. Experiment 3: Is Method 3 more effective than Method 2?

1) Purpose: This experiment investigates an attack whereby a simple transformation (such as prepending a short piece of text) is applied to two files. After this step, the resulting files undergo ZIP compression and the resulting file size is compared. Is it possible to improve the success of ZIP detection of file variations that may represent fingerprints?

2) Method: The PDF files used in the earlier experiments were reused here, and variant files were generated as before. This time, a short phrase was edited into the beginning of the two files being compared. The phrases used in these tests were: *“PANCAKES”*, *“12345”*, *“BELL”*, *“LEE”*, and *“x”*. Informal tests suggested that the exact choice of phrases used is not significant, as long as a few different phrases are chosen. Notice that even a non-technical user could add such short phrases very quickly and easily with the GUI-based text editors available from a basic installation of Windows or Mac OS (Wordpad, TextEdit).

5 paired copies of the two varying files were produced. Each pair of copies had the same fixed short phrase prepended. Each file was then ZIP compressed. The resulting file sizes in bytes were compared. The final percentage of non-detected variations was the number of variations that were not detected using any of the 5 phrases. In total, 10 PDF files were tested, with 10 values of n and 100 different variations, and 5 prepended phrases, yielding 50,000 experiments.

3) Results: Table III shows the results of the experiment. The mean detection rate achieved by Method 2 is shown again in this table, to allow results to be conveniently compared.

Again, these experimental results measure the percentage of cases where dissimilarity could be detected through a change in the observable file size. 0% indicates that no variations were detected; 100% indicates that all variant files were detected with at least one of the five test phrases.

K. *Discussion of results*

- Method 3 offers a uniformly improved success rate in detection, compared with Method 2. Particularly, in the cases with $n = 32$, Method 3 was found to be five times less likely to produce a false negative during detection than Method 2, on average. Method 2 was already 95% successful in detecting variations of that size.
- As before, at some point all 100 variant files of all 10 PDFs were detectable, once the degree of variation became sufficiently large. This limit was reached sooner than in Experiment 1. This suggests that Method 3 is even more effective than Method 2.
- Method 3 provides an opportunity to foil 'lucky' variations that evaded Method 2.
- Even when only 1 bit varied in 8 million bits, Method 3 detected the variation in almost 50% of cases, on average.
- In the best cases of uncompressed media, Method 3 was almost 100% successful at detecting any variation between two files whatsoever; even a single bit.
- In the worst case, with very poorly compressible data, Method 3 had a better than 50% chance of detecting changes consisting of more than 8 bits between two files.
- No false positives were detected.
- It is still possible that some tiny variations in poorly compressible files can go undetected despite the use of several test phrases. This leaves an interesting target for future study and improvement of this approach.

In general, Method 3 was found here to be an extremely effective approach for detecting subtle bit-level variations between files, except in cases where a total of 4 bits or fewer varied between two 1MB files, and where the file was essentially entirely composed of poorly compressible data.

L. *Fingerprints and variations in the real world*

All three methods will necessarily produce a type of false positive, whenever two documents are presented with minor variations that do *not* represent the deliberate introduction of a digital fingerprint, but instead are due to some accidental change. It is impossible for the restricted whistleblower to be certain about which detected variations represent *deliberately* introduced fingerprints, and which are *accidental* changes introduced in handling the file.

Nonetheless, today's accidentally introduced differences become tomorrow's digital fingerprint which retrospectively identifies the whistleblower. Consequently there is arguably no such thing as a *false positive* in the realm of cautious digital whistleblowing - any variation whatsoever between two files that are meant to be identical, might represent either a deliberately introduced digital fingerprint now, or an accidentally introduced change that is used as an ad hoc digital fingerprint in the future. In either case, variations between supposedly identical files are a serious risk to anonymity when leaking a document. Whistleblowers should therefore be careful to avoid accidentally modifying the data of any of the files they intend to leak, for this reason.

VI. Summary and Conclusions

This paper has presented the problem of digital whistleblowing, where a document is to be leaked that may contain an invisible embedded digital fingerprint, in environments that are very restricted in terms of the technical skills of whistleblower, freedom of user behaviour, permitted communication, and available software tools.

The digital fingerprint detection methods presented here represent simple yet novel and effective contributions to the defence of naive digital whistleblowers in restricted environments. The idea of using compressed file size information to allow indirect comparison of the data in two very similar files may be novel and useful in other situations besides whistleblowing; for example, when comparison of similar, isolated files is desired with minimal communication, but where traditional software for calculating checksums or carrying out byte-by-byte comparison is for some reason inconvenient.

The experimental results presented here demonstrate that these methods are practical and beneficial under the assumed conditions of this paper. In particular, these methods and experimental results present a challenge to fingerprinting tools that claim to be unnoticeable to ordinary, non-technical users without access to specialised tools for analysing files. However, these techniques do not guarantee safety absolutely. Future research into the *whistleblower-defending problem* should therefore aim to find further methods, suitable for ordinary people, that are even more successful and easy to use for the purpose of protecting everyday whistleblowers against digital fingerprints.

The author recommends that all whistleblowers - and those who rely upon them - should become familiar with these methods and should try to invent better techniques.

VII. Closing comment from the author

I have chosen to publish this paper in an open access journal, because I feel members of the public should not be restricted in their ability to access knowledge that may protect them from reprisals. I encourage any future authors who address the whistleblower-defending problem to consider publishing in the same manner.

References

AICPA. Anonymous submission of suspected wrongdoing (whistleblowers). *American Institute of Certified Public Accountants*, 2005. URL http://www.globalcompliance.com/pdf/AICPA_whistleblower.pdf.

N. Anthapadmanabhan, A. Barg, and I. Dumer. On the fingerprinting capacity under the marking assumption. *IEEE Transactions on Information Theory*, 54(6):2678–2689, June 2008. URL http://arxiv.org/abs/cs/0612073v3.

G. R. Blakley, C. Meadows, and G. B. Purdy. Fingerprinting long forgiving messages. In *Advances in cryptology—CRYPTO 85*, volume 218 of *LNCS*, pages 180–189, New York, NY, USA, 1986. Springer-Verlag New York, Inc. ISBN 0-387-16463-4. URL http://portal.acm.org/citation.cfm?id=25397#.

D. Boneh and J. Shaw. Collusion secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, pages 452–465, 1995. URL http://portal.acm.org/citation.cfm?id=706008.

J. T. Brassil, S. Low, and N. F. Maxemchuk. Electronic marking and identification techniques to discourage document copying. In *IEEE Journal on Selected Areas in Communications*, pages 1278–1287, 1994. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.104.1989.

J. T. Brassil, S. Low, and N. F. Maxemchuk. Copyright protection for the electronic distribution of text documents. In *Proceedings of the IEEE*, pages 1181–1196, 1999. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=771071.

B. Chor, M. Naor, and B. Pinkas. Tracing traitors. In *Advances in Cryptology, Proceedings of CRYPTO '94*, volume 839 of *Lectures Notes in Computer Science*, pages 257–270. Springer-Verlag, 1994. URL http://www.cs.ucla.edu/~miodrag/cs259-security/chor94tracing.pdf.

I. J. Cox and M. L. Miller. Electronic watermarking: the first 50 years. In *IEEE Workshop on MultiMedia Signal Processing*, pages 225–230, 2001. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=962738.

I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon. A secure, robust watermark for multimedia. In *Workshop on Information Hiding, Newton Institute, Univ. of Cambridge*, 1996. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.1387.

I. J. Cox, J. Kilian, T. Leightonz, and T. Shamoony. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997. URL http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.118.9444.

P. Deutch. DEFLATE compressed data format specification version 1.3. IETF RFC 1951, May 1996. URL http://tools.ietf.org/html/rfc1951.

J. J. Eggers, J. K. Su, and B. Girod. Asymmetric watermarking schemes. In *in Sicherheit in Mediendaten, GMD Jahrestagung, Proceedings*. Springer Verlag, 2000. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.8479.

Ernst and Young. Fraud: The unmanaged risk: 8th global survey. *Ernst and Young Global Investigations & Dispute Advisory Services*, 2002. URL http://www.your-call.com.au/information/documents/EY8thGlobalSurvey2003.pdf.

A. Feldspar. An explanation of the 'deflate' algorithm. http://zlib.net/feldspar.html, August 1997. URL http://www.gzip.org/deflate.html.

A. Fiat and T. Tassa. Dynamic traitor tracing. In *CRYPTO' 99*, volume 1666 of *LNCS*, pages 354–371. Springer, 1999. URL http://www.springerlink.com/content/7j3aldmwjvg41pk9/.

N. Heintze. Scalable document fingerprinting. In *In Proc. USENIX Workshop on Electronic Commerce*, 1996. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.8072.

D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4051119.

L. Jong-Hyeon. *Fingerprinting (Information Hiding Techniques for Steganography and Digital Watermarking)*, chapter 8, pages 175–189. Artech, 2000.

S. Katzenbeisser and A. P. Fabien, editors. *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech, 2000. URL http://www.amazon.com/dp/1580530354.

G. Kessler. An overview of steganography for the computer forensics examiner. *Forensic Science Communications*, 6(3), 2004. URL http://www.fbi.gov/about-us/lab/forensic-science-communications/fsc/july2004/research/2004_03_research01.htm/.

J.-H. Lee, T.-H. Lim, K.-W. Kim, and S.-U. Shin. A new fingerprinting codes for multimedia contents. In *Adv. in Multimedia Modelling*, volume 4352 of *LNCS*, pages 510–519. Springer, 2006. URL http://www.springerlink.com/content/d9888w52535613v2/.

Y.-T. Lin and J.-L. Wu. Content adaptive watermarking for multimedia fingerprinting. *Computer Standards and Interfaces*, 30:271–287, 2008. ISSN 0920-5489. URL http://portal.acm.org/citation.cfm?id=1367236#.

S. Low and N. Maxemchuk. Performance comparison of two text marking methods. *IEEE Journal on Selected Areas in Communications*, 16(4):561–572, 1998. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.25.1067.

S. H. Low and N. F. Maxemchuk. Modeling cryptographic protocols and their collusion analysis. In *Proc. International Workshop on Information Hiding*, pages 169–184, London, UK, 1996. Springer. ISBN 3-540-61996-8. URL http://portal.acm.org/citation.cfm?id=731516.

D. MacKenzie, P. Eggert, and R. Stallman. Comparing and merging files with gnu diff and patch. Technical report, Network Theory Ltd., 2003. URL http://digipen2.xmmg.com/dpweb/docs/DiffAndPatch.pdf.

N. F. Maxemchuk. Electronic document distribution. *ATT Technical Journal*, pages 73–80, September 1994. URL http://www.ee.columbia.edu/~nick/ref.1317.ps.

N. Nikolaidis and I. Pitas. Robust image watermarking in the spatial domain. In *Signal Processing*, pages 385–403, 1998. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.7820.

J. H. Park, S. E. Jeong, and C. S. Kim. Robust and fragile watermarking techniques for documents using bidirectional diagonal profiles. In *ICICS 2001*, volume 2229 of *Information and Communications Security*, pages 483–494. Springer Berlin / Heidelberg, 2001. URL http://www.springerlink.com/content/0tgk677f4v6ydg8a/.

B. Pfitzmann. Trials of traced traitors (extended abstract). In *Proceedings of the First International Workshop on Information Hiding*, volume 1174 of *LNCS*, pages 49 – 64, 1996.

URL http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.3740.

B. Pfitzmann and M. Schunter. Asymmetric fingerprinting (extended abstract). In *EUROCRYPT '96*, volume 1070 of *LNCS*, pages 84–95. Springer, 1996. URL http://en.scientificcommons.org/42857352.

B. Pfitzmann and M. Waidner. Anonymous fingerprinting. *LNCS*, 1233:88–104, 1997. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.4022.

N. Provos. Stegdetect, 2004. URL http://www.outguess.org/detection.php.

N. Provos and P. Honeyman. Hide and seek: An introduction to steganography. *IEEE Security and Privacy*, 1(3), 2003. URL http://dx.doi.org/10.1109/MSECP.2003.1203220.

J. D. Ratley. 2008 report to the nation on occupational fraud and abuse. *Association of Certified Fraud Examiners*, 2008. URL www.acfe.com/documents/2008-rttn.pdf.

R. Rivest. The MD5 Message-Digest Algorithm. Technical Report RFC1321, MIT Laboratory for Computer Science and RSA Data Security, Inc., April 1992. URL http://tools.ietf.org/html/rfc1321.

Steganography Analysis and Research Center. Steganography Analyzer Artifact Scanner (StegAlyzerAS), May 2011. URL http://www.sarc-wv.com/products/stegalyzeras/.

Y.-L. Tang and C.-P. Wang. A robust watermarking algorithm based on salient image features. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2008. URL http://www.computer.org/portal/web/csdl/doi?doc=doi/10.1109/IIH-MSP.2008.65.

A. Tirkel, G. Ranking, R. van Schyndel, W. Ho, N. Mee, and C. Osborne. Electronic watermark. In *Dicta-93*, pages 666–672, 1993. URL http://goanna.cs.rmit.edu.au/~ronvs/papers/DICTA93.PDF.

W. Trappe, M. Wu, and K. Liu. Collusion-resistant fingerprinting for multimedia. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pages 3309–3312, 2002. URL http://sig.umd.edu/publications/Trappe_ICASSP_200205.pdf.

W. Trappe, M. Wu, Z. Wang, and K. Liu. Anti-collusion fingerprinting for multimedia. In *IEEE Transactions on Signal Processing*, volume 51, pages 1069–1087, 2003. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.7136.

U.S. G.P.O. *Sarbanes-Oxley Act of 2002*, volume H.R. 3763. U.S. G.P.O., 2002. URL http://www.sec.gov/about/laws/soa2002.pdf.

U.S. G.P.O. *Whistleblower Protection Act*, volume H.R. 985. U.S. G.P.O., 2007. URL http://www.usda.gov/oig/webdocs/whistle1989.pdf.

N. R. Wagner. Fingerprinting. In *IEEE Symposium on Security and Privacy*, page 18. IEEE Computer Society, 1983. URL http://cs.utsa.edu/~wagner/pubs/finger/fing2.pdf.

Z. J. Wang, M. Wu, W. Trappe, and K. J. R. Liu. Group-oriented fingerprinting for multimedia forensics. *EURASIP J. Appl. Signal Process.*, 2004(1):2153–2173, 2004. ISSN 1110-8657. URL http://sig.umd.edu/publications/wang_group_200411.pdf.

Wikipedia. MD5, May 2011. URL http://en.wikipedia.org/wiki/MD5.

M. Wu and B. Liu. *Multimedia Data Hiding*. Springer-Verlag, 2002. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.93.6348.

M. Wu, W. Trappe, Z. Wang, and K. Liu. Collision-resistant fingerprinting for multimedia. *IEEE Signal Processing Magazine*, 21(2):15–27, Mar 2004. URL http://www.ece.umd.edu/~minwu/public_paper/Jnl/0403FPcollusion_IEEEfinal_SPM.pdf.

J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23:337–343, 1977. URL http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.118.8921.