# PLAGIARISM FROM A DIGITAL FORENSICS PERSPECTIVE

Clare S. Johnson, Ross Davies

**Abstract:** Plagiarism and contract cheating are serious academic issues that 'undermine the integrity of education' (Bretag, 2013). There are a number of tools that can help assessors detect plagiarism – particularly where text has been copied and pasted: Turnitin (`https://www.turnitin.com`), PlagScan (`https://www.plagscan.com`) and Urkund (`https://www.urkund.com`) are examples of such tools. The providers of these tools are also developing authorship tools that use stylometrics and linguistics to determine matches between authors (whether the submitting author, or a third party). It is also possible for an assessor to copy passages of text and paste them into a Google search (or similar) with quotes surrounding the passage to see if there are any immediate online matches. In a previous paper *Using digital forensic techniques to identify contract cheating: A case study* (Johnson & Davies, 2020), the authors began using digital forensic techniques to see if it was possible to detect contract cheating. In that paper, consideration was given to how forensic techniques allow review of document edits through examining the Open Office extensible markup language (OOXML) format of the document. This paper seeks to extend that research by further exploration of the OOXML format to establish whether forensic artefacts can be found to indicate that work has been copied and pasted from online sources. A number of sample documents were created by the authors and the xml analysed. Whilst there were some indicators to suggest work had been copied and pasted, more analysis is required to develop the techniques into a more reliable tool.

**Key words:** Cheating, Copying, Detection, Forensics, OOXML, Plagiarism

## Literature Review

The ease of copying and pasting information from the Internet can be very tempting for students, particularly when they are struggling with an essay, or when deadlines are tight and they are under pressure. According to the website Plagiarism.org (plagiarism.org, 2017), plagiarism is an act of fraud, which involves 'both stealing someone else's work and lying about it afterwards' – in other words, submitting work as your own when it has actually been copied from another source. Extensive surveys by McCabe (2001) found that 36% of students admitted to "paraphrasing/copying a few sentences from Internet source without footnoting it" (plagiarism.org, 2017; McCabe 2001). Bretag's review (2014) of numerous surveys notes that the key findings have been that 'breaches of academic integrity are rife in colleges and universities around the world', and that little has changed since the early surveys of Bowers (1964) and McCabe. In 2016, an article written for the Times newspaper, based on Freedom of Information Requests, found that almost 50,000 students at British universities had been caught cheating in the previous three years, with only 362 students being withdrawn from their courses as a result (Mostrous & Kenber, 2016).

There are serious implications arising from plagiarism too. Foltynek, Meuschke & Gipp, (2019), explain that for the academic student, plagiarism is 'detrimental to

DOI: https://doi.org/10.1118/978-80-7509-772-9-0078

competence acquisition and assessment', which in turn could lead to professionals who appear to be qualified, but who in fact have achieved their qualifications based on someone else's work; whilst plagiarised research papers can 'impede the scientific process . . . by distorting the mechanisms for tracing and correcting results'. In addition, McCabe, Trevino and Butterfield (1996) discuss how strong ethics codes (or 'honor codes') in college students are shown to reduce the likelihood of self-reported unethical behaviour in the workplace, which could suggest that students who engage in unethical behavior during their studies may later bring this behaviour into their workplaces.

There is not a great deal of research that looks into the use of digital forensics to establish ownership of student submissions. Tools such as Turnitin (`https://www.turnitin.com`), PlagScan (`https://www.plagscan.com`) and Urkund (`https://www.urkund.com`) are very much text matching tools, which scan archives for passages of text that appear in student submissions. By changing one or two words, or switching phrases around, students can outwit the mechanisms used by these tools and achieve low scores, which then in turn fail to alert assessors that potential plagiarism has occurred. Developments are in place to improve these tools by adding in authorship tools and stylometrics, and these will certainly help in flagging potential plagiarism or outsourced work.

The authors of this paper both have experience of working in the cyber security teaching department of a Higher Education Institution, and have taught digital forensics and cyber security. As such, they wondered whether the techniques used to identify ownership of certain files involved in criminal investigations could potentially be applied to student submissions to establish ownership and originality of work.

Research into the application of digital forensics using the OOXML approach include the use of digital forensic techniques for identifying copyright issues (Fu, Sun, Liu & Li, 2011) and intellectual property (Jeong & Lee, 2017), though the latter requires access to the entire file system of the computer where the document was created, and Xiang, Sun, Liao & Wang (2016) who discuss the use of extensible markup language for transmission of secret information. (Jeong & Lee, 2017). As (Didriksen, 2014) notes: "it is desirable to connect the actions performed, e.g. editing the document, to a specific physical person or several people" when carrying out a digital forensics investigation, as this permits investigators to attribute certain actions to specific users. The authors of this paper wonder if similar forensic investigation techniques can be useful in establishing the originality of work submitted by a student.

## Research aim and objectives

This object of this paper is to explore the use of OOXML to see if there are other flags or features that might raise suspicion that a piece of work has not been created in an authentic way and hence may be plagiarised. Specifically, the research aims to:

- Describe various stylistics features of OOXML;
- Analyse which features of OOXML may be useful in determining the authenticity of a document;

- Determine the extent to which forensic analysis of these features can help determine originality.

# Methodology

## Device and Document Specification

After an initial pilot study, comprising one plagiarised document (created by the authors) and one fully original document, a new batch of ten documents, referred to as 'sample files', were created. Three documents were created on Device A, one on Device B, four on Device C and two on Device D. The specifications for each device (operating system and software version) are detailed below [Table 1].

Table 1
*Devices and documents*

| Operating System | Word Version | Documents created |
|---|---|---|
| Device A:<br>MacOS High Sierra v 10.13.6 | Microsoft Word for Mac v 16.34 (20020900) | • Human Computer Interface Library<br>• Plagiarism Essay Wikipedia<br>• Python Programming Essay Stack Overflow<br>• Raspberry Pi Essay Github |
| Device B: Windows 10 Education v 1803 | 2019 MSO (16.10348.20020) | • Red Team Blue Team |
| Device C: Windows 10 Education v 1903 | 2016 (16.0.4954.1000) | • Securing the Internet of Things<br>• Computer Programming Wikipedia<br>• Social Engineering Authentic |
| Device D: Windows 10 Education v 1903 | Microsoft Word for Office 365 (16.0.11727.20222) | • Blockchain<br>• Online Discussion Fora |

In each case, some original text was typed into the document and the document was saved. Information from another source was then obtained through Internet searches on various websites as indicated later in this study. This information was copied and pasted directly into each original document, formatted to match the original text and resaved. This is similar to the actions taken by students who copy and paste information from other sources – whether citing them correctly or otherwise. The method used for reformatting to match the original document varied from sample to sample, with some samples using the Format Painter (sometimes known as Paste Format) tool, and others using manual font adjustments. Copying and pasting text from multiple sources is sometimes known as 'patchworking', where students take passages of text from the Internet and build them into a submission, without giving adequate credit (Kumar, P. M., Priya, N. S., Musalaiah, S., & Nagasree, 2014). In the sample files created for this study, the source of the copied material is *not* included in the sample files, but is included as a subsection of the references for this article. One of the documents ('Social Engineering Authentic') uses original text, supplemented by a minimal amount of information sourced from the Internet that has been paraphrased or summarised, to replicate the typical actions of a non-plagiarising student.

Once each document was saved, it was then converted to a zip file using the process specified below, and the document.xml file reviewed in Notepad++ and Chrome to establish whether forensic artefacts relating to the process of copying and pasting could be found. If so, these could potentially be considered flags for plagiarised work.

## OOXML format

As discussed in the previous paper by Johnson & Davies (2020), Microsoft Word uses 'Office Open XML Format' (OOXML), where a document is created from a combination of other underlying documents. Much like a film is made up of many scenes, with music, special effects and credits added, and finally packaged up into a single item, a Word document (docx) is made up of several other files, compressed into a single package.

The Open XML format has been around since Microsoft Office 2007 and was designed to bring several benefits to individuals, organisations and developers. These benefits include improved damage recovery because the various components of each document are stored separately, meaning that if one component is damaged it may still be possible to open the file; better privacy and control over personal information, because sensitive information can be more easily identified and thus removed if required, and more compact: it is this feature that we can take advantage of when carrying out a digital analysis of the file (Microsoft, 2019).

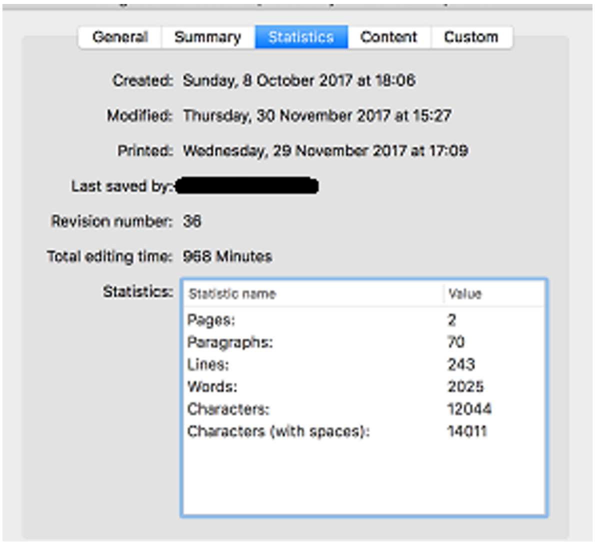
It is simple to review the document properties of a file when opened. This information can be found under the **File** menu by selecting **Properties** and then the **Statistics** tab (Fig. 1) (depending on Word version). This information can be useful, but is not always reliable. Instead, looking inside the packaged contents of this document can reveal much more interesting data about the file and the way it was created.

To look inside the compressed docx file (or package), it first needs to be decompressed by changing the extension of the file from.docx to .zip, and then choosing Extract, or Unzip (depending on your system). Opening the folder that is created then reveals a series of subfolders: rels; docProps; word and a single file [ContentTypes].xml.

The **[Content_Types].xml** at the root of the folder contains a list of the content types of the parts within the package. The **_rels** folder tells Word how the parts relate to each other and to resources outside of the package.

Within the **word** folder, we find the following content as a minimum: _rels, theme (folders); document.xml; settings.xml; styles.xml. The file containing most of the content is the **document.xml** file, and this is the file focused on for this paper. The **document.xml** file is the main xml file for the document and includes the document's content and run identifiers. These run identifiers (RsiD tags) indicate how the document was built by placing each and every edit inside a tag, or 'run'. In the previous paper by the authors, a detailed review of the run identifies was carried out.

The samples created for this research are detailed as follows:



Created: Sunday, 8 October 2017 at 18:06

Modified: Thursday, 30 November 2017 at 15:27

Printed: Wednesday, 29 November 2017 at 17:09

Last saved by: [redacted]

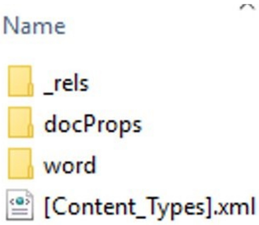
Revision number: 36

Total editing time: 968 Minutes

Statistics:

| Statistic name | Value |
|---|---|
| Pages: | 2 |
| Paragraphs: | 70 |
| Lines: | 243 |
| Words: | 2025 |
| Characters: | 12044 |
| Characters (with spaces): | 14011 |

*Figure 1.* Document Properties panel


Name

_rels
docProps
word
[Content_Types].xml

*Figure 2.* Decompressed docx file

# Results and Discussion

## Flags for plagiarised work

Inspection of the document.xml files for each sample file showed a number of forensic artefacts that were investigated further to determine whether they could be considered flags for copied and pasted text.

Code inside the document.xml file tells Word how to render the document when displaying it on screen. Styling is defined within a w: namespace, which is developed by adding a relevant element. For example w:document tells Word that it is looking at a Word document; whilst w:body indicates that what follows is the body text of the document. A genuinely created Word document yields a number of typical xml instructions, but in the samples some anomalies were detected. In the mainly authentic sample (‘Social Engineering Authentic’), there is no appearance of the w:rFont element

Table 2
*Documents created and device information*

| Document Name | Device | Source used | Notes |
|---|---|---|---|
| Human Computer Interface Library | Device A (Mac) | Library – online journal articles | Two sources used |
| Plagiarism Essay Wikipedia | Device A (Mac) | Wikipedia | |
| Python Programming Essay Stack Overflow | Device A (Mac) | Stack Overflow | |
| Raspberry Pi Essay Github | Device A (Mac) | Github | |
| Securing the Internet of Things (IoT) | Device B (Windows) | Online journal article and blog post | Two sources used |
| Social Engineering Authentic | Device B (Windows) | | Text copied for reference only, heavily edited or paraphrased |
| Computer Programming Wikipedia | Device B (Windows) | Wikipedia | |
| Red Team Blue Team | Device C (Windows) | Wikipedia and two online blog posts | Three sources used |
| Blockchain | Device D (Windows) | Web pages resulting from Google search | Two sources used |
| Online Discussion Fora | Device D (Windows) | Website and online journal article | Two sources used |

within the file. Even when the w:rFont element appears in other original examples tested by the researchers, it would be to specify a single font attribute, e.g. w:rFonts w:eastAsia="Times New Roman", unless the passage where it appears is part of a field entry (e.g. Table of Contents entry) or the font has been changed (i.e. not default).

However, the 'Python Programming' file includes multiple font attributes as shown:

```
<w:rFonts w:ascii="inherit" w:eastAsia="Times New Roman" w:hAnsi="inherit"
w:cs="Consolas"/>
```

This is similar for the 'Securing the Internet of Things (IoT)' file:

```
<w:rFonts w:asciiTheme="minorHAnsi" w:hAnsiTheme="minorHAnsi"
w:cstheme="minorHAnsi"/>
```

Why this occurs is not clear, but it is possible that accessing the text across several versions (i.e. initially in the original online version, then additionally through copying into the Word document) may result in a number of font specifications, each of which relates to a version of the text (HTML or CSS for the web, Microsoft Word for the Word version). To test this theory, a very simple document was created in Word called 'Test.docx' with one line of text typed directly into the document, and a second line of text copied directly from the Internet, with no formatting applied. This shows a specification of the font in the copied section, which is not required in the authentic typed-in text (as it simply uses the default font). The Discussion Fora also demonstrates a number of `<w:rFont>` elements. Referring to the Open Office XML glossary indicates that the attributes applied in the rFont element are used to allow for the display of different subsets of Unicode characters (i.e. not those from the default set), such as Asian characters or Arabic text. It is important, therefore to consider that these attributes may appear through conversion from one language (perhaps the native language of the student) into English, and not because of academic misconduct.

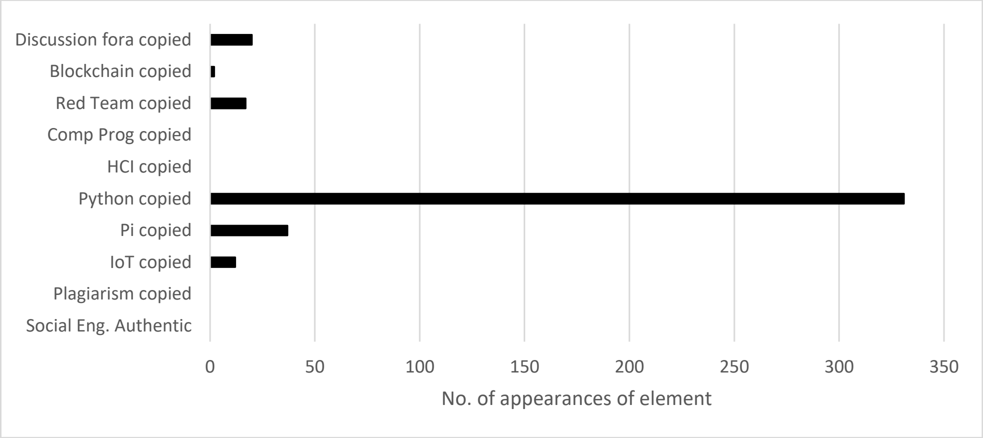


*Figure 3.* Frequency of `w:rFonts` element in sample files

Other formatting elements can also be seen across the samples created as shown below and are discussed later in the study:

```
<w:rPr>
      <w:rFonts w:ascii=”Verdana” w:hAnsi=”Verdana”/>
      <w:color w:val=”000000”/>
      <w:sz w:val=”23”/>
      <w:szCs w:val=”23”/>
      <w:shd w:val=”clear” w:color=”auto” w:fill=”FFFFFF”/>
</w:rPr>
```

When the copied text is formatted using the Format Painter tool and resaved, these formatting elements are typically removed. The ‘Programming Wikipedia’ file, which contains text copied from Wikipedia and reformatted to match the font of the original document does not include any rFonts specifications at all, most likely because of the use of this tool.

The `<w:rPr>` element defines the run properties for a particular run of edits, including attributes for font face, size and language. In the samples containing copied text there are some interesting features of this `<w:rPr>` element. In this example, taken from the ‘Red Team Blue Team’ file, we can see that several attributes are defined within the `<w:rPr>` element. However, this is not followed by any text to appear on screen (w:t), but is, in fact, followed by another `<w:rPr>` element, suggesting that the attributes are redundant and possibly a relic of previous formatting:

```
<w:rPr>
```

```
      <w:rFonts w:eastAsia=”Times New Roman” w:cstheme=”minorHAnsi”/>
      <w:sz w:val=”24”/>
      <w:szCs w:val=”24”/>
      <w:lang w:eastAsia=”en-GB”/>
</w:rPr>
```

This feature is present in several of the sample files, including ‘Securing the Internet of Things’, ‘Raspberry Pi’ and ‘Python Programming’. In fact, the ‘Python Programming’ file includes a huge number of rPr elements compared to the other documents, which may be because it contains code snippets.

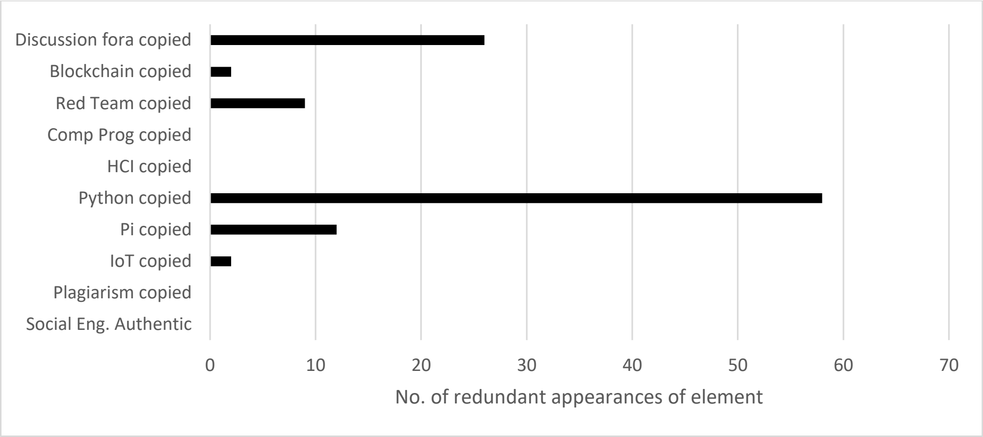


*Figure 4.* Frequency of redundant rPr elements

There were several elements that were expected to be found in the documents containing copied text from preliminary work, but these did not appear in significant numbers during the testing for this paper. For example, the font element `<w:sz>` (which relates to font size) did not appear in all the copied and pasted examples, but appeared significantly in four out of the seven plagiarised works:

The `<w:shd>` element denoting that a shadow (background) has been applied to the run and which the authors found this in the pilot phase of the study, only appeared in documents where reformatting had been done manually (by highlighting the passage, and applying the correct font using the Font tool) as in ‘Securing the Internet of Things’, ‘Blockchain’ and ‘Discussion Fora’. This could correlate with copied and pasted text, as text created within Word itself would already have a white or null background shadow, and therefore the appearance of this command suggests that the text has come from elsewhere. This should have been considered a flag for copied work that had to be reformatted to remove unwanted attributes and was also seen in the ‘Test.docs’ file before the Format Painter was used to match the formatting of the original document. Text that is reformatted using the Format Painter does not bear this characteristic.

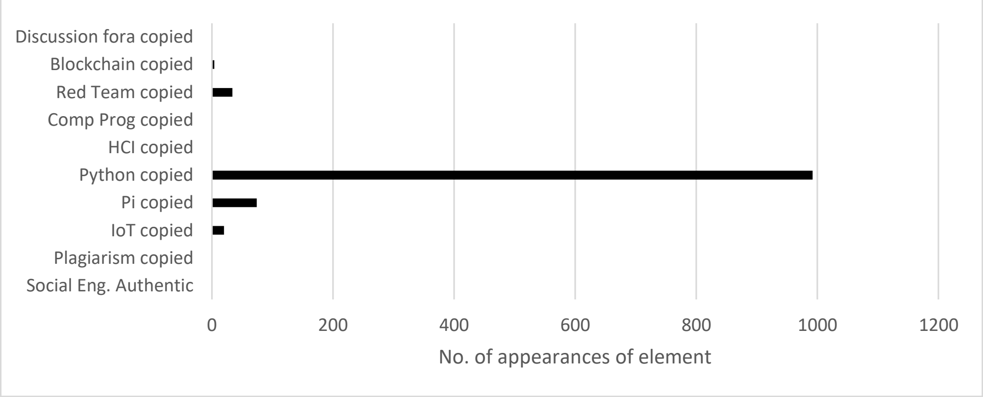


*Figure 5.* Frequency of w:sz element in sample files

## Other interesting findings

Other tags and elements within the pilot study reviewed yielded interesting appearances of `<w:NormalWeb>` and `<w:webHidden>`, neither of which appeared in the original work during the pilot. However, the appearance of `<w:NormalWeb>` only appears in the 'Securing the Internet of Things' and 'Blockchain' files in the full study, and `<w:webHidden>` does not occur at all.

It is worth commenting on two of the samples analysed, namely 'Python' and 'Pi' as these clearly demonstrate higher incidents of most of the elements and attributes discussed above. Carrying out a count on the number of 'words' in the associated document.xml files (whether code or actual text) reveals that files that copy 'code' from the Internet demonstrate a much higher ratio of xml words to the number of text words in the original.docx file. Whilst most of the samples demonstrate between 4 and 11 times the number of xml words to original text words, the 'Python' and 'Pi' samples show 98 times and 219 times more words respectively. This may be because of the extended formatting required in these documents, but is more likely as a combined result of the extended formatting found on the Internet version of the file which need removing, plus the formatting within the document itself. The relevance of this word count would be interesting to review in further detail, particularly in subject areas which require the use of code, to establish whether the formatting required for code naturally results in higher xml word counts, or whether this is solely because of the reformatting requirements for code copied from the Internet.

## Limitations

As with many digital forensics techniques, these flag can only act as indicators. There may be genuine reasons why a document includes such flags, and it is to be expected that some information will come from online sources as part of the proper literature review, though of course these sources should be paraphrased, summarised or extended and cited accurately. The methods described in this study are also dependent on the

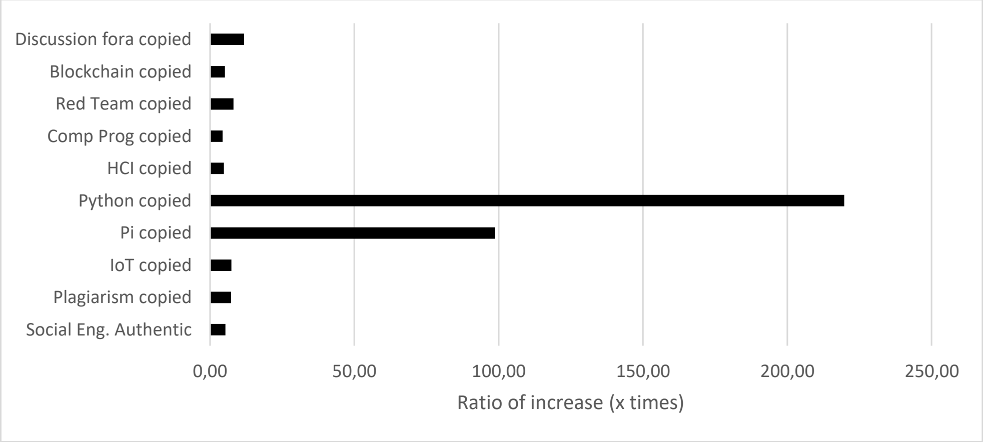


*Figure 6.* Number of xml words in relation to document word count

student submitting the assignment as a word.docx file, and not a PDF or other format, which have not yet been investigated.

## Conclusions and Further Work

It is often possible to identify plagiarism through the use of text matching software, or by using search engines to find suspicious paragraph of text. However, by changing a single word, or by patchworking, students are able to outsmart tools like Turnitin and PlagScan, and this can render any online searches for similar passages by the assessor unsuccessful. Reviewing the xml of the submission does not enable an assessor to categorically state whether work is plagiarised, but it is another option in the toolkit for highlighting flags which may be indicators of plagiarised work. The authors believe that there is much more that can be done in this area, perhaps developing tools which review the xml format in greater detail, and also to clarify how these elements and attributes are applied during the copy and paste process in more detail. Furthermore, the elements need to be reviewed in a more holistic way, as singling out elements and attributes in isolation from the final, fully rendered document makes them less meaningful. It would also be useful to review documents created in Google Docs and Libre Office, and those in PDF format, to see if these highlight any forensic artefacts of interest. Finally, ethical approval for the analysis of real examples of student work should be gained, as these will provide the richest source of data.

## References

Bowers, W. J. (1964). Student dishonesty and its control in college. New York: Bureau of Applied Social Research, Columbia University.

Bretag, T. et al. (2014) Teach us how to do it properly! An Australian academic integrity student survey, *Studies in Higher Education*, 39(7), pp. 1150–1169. doi: 10.1080/03075079.2013.777406.

Foltynek, T., Meuschke, N., & Gipp, B. (2019). Academic Plagiarism Detection: A Systematic Literature Review, *ACM Computer Surveys,* 12 doi: 10.1145/3345317, accessed 23/03/20

Fu, Z., Sun, X., Liu, Y., & Li, B. (2011). Forensic investigation of OOXML format documents. *Digital Investigation, 8*(1), 44–55. doi:10.1016/j.diin.2011.04.001

Jeong, D., & Lee, S. (2017). Study on the tracking revision history of MS Word files for forensic investigation. *Digital Investigation, 23,* 3–10. doi:10.1016/j.diin.2017.08.003

Johnson, C., Davies, R. (2020). Using Digital Forensic Techniques to Identify Contract Cheating: A Case Study. Journal of Academic Ethics. https://doi.org/10.1007/s10805-019-09358-w

McCabe, D., Trevino, L., & Butterfield, K., (1996). The Influence of Collegiate and Corporat Codes of Conduct on Ethics-Related Behavior in the Workplace. *Business Ethics Quarterly*, 6(4), pp. 461–476. https://doi.org/10.2307/3857499

Mccabe, D., Trevino, L., & Butterfield, K., (2001). Cheating in academic institutions: A decade of research. Ethics & Behavior, 11(3), pp. 219–232.

Microsoft. (2019). Open XML Formats and file name extension. Retrieved from: https://support.office.com/en-gb/article/open-xml-formats-and-file-name-extensions-5200d93c-3449-4380-8e11-31ef14555b18, accessed 30/11/19

Mostrous, A., & Kenber, B. (2016). Universities face student cheating crisis. *The Times*, available at: https://www.thetimes.co.uk/article/universities-face-student-cheating-crisis-9jt6ncd9vz7

Plagiarism.org (2017). What is Plagiarism. Available at: https://www.plagiarism.org/article/what-is-plagiarism, accessed 28/02/20

Xiang, L., Sun, C., Liao, N., & Wang, W. (2016). A Characteristic-Preserving Steganographic Method based on Revision Identifiers. *International Journal of Multimedia and Ubiquitous Engineering, 11*(9), 29–38.

## References for sample essays

### Plagiarism Essay

Wikipedia (2020). Plagiarism, Available at: https://en.wikipedia.org/wiki/Plagiarism, accessed 20/02/20

### Python Essay

Stack Overflow (2020). Creating a traffic light using Python, Available at: https://stackoverflow.com/questions/24588406/creating-a-traffic-light-using-python, accessed 20/02/20

### Raspberry Pi Essay

Github (2020). Pi-hole / setup.py, Available at: https://github.com/pi-hole/pi-hole/blob/master/setup.py, accessed 20/02/20

### Human Computer Interface Essay

Y. Ma, Z. Mao, W. Jia, C. Li, J. Yang and M. Sun (2011). Magnetic Hand Tracking for Human-Computer Interface. In *IEEE Transactions on Magnetics*, vol. 47, no. 5, pp. 970–973

Barszap, A. G., Skavhaug, I., Joshi, S. S., (2016). Effects of muscle fatigue on the usability of a myoelectric human-computer interface, *Human Movement Science,* vol. 49, pp. 225–238, available at: https://10.1016/j.humov.2016.06.009, accessed 20/02/20.

### IoT Essay

IEEE (2018). Internet of Things Security: Is Anything New? In *IEEE Security & Privacy*, vol. 16, no. 5, pp. 3–5, September/October 2018, available at: https://doi.org/10.1109/MSP.2018.3761715, accessed 19/02/2020.

Intellectsoft (2019). Top 10 Biggest IoT Security Issues, available at https://www.intellectsoft.net/blog/biggest-iot-security-issues/, accessed 19/02/2020

### Red Team Blue Team Essay

Wikipedia (2020). Red Team, Available at: https://en.wikipedia.org/wiki/Red_team, accessed 28/02/20).

SecurityTrails (2018). Cybersecurity Red Team Versus Blue Team – Main Differences Explained, available at: https://securitytrails.com/blog/cybersecurity-red-blue-team, accessed 28/02/20

EC Council (n.d.). Red Team vs Blue Team, available at: https://blog.eccouncil.org/red-team-vs-blue-team/, accessed 28/02/20.

### Social Engineering Essay

Norton (2020). What is social engineering? Tips to help avoid becoming a victim. Available at: https://us.norton.com/internetsecurity-emerging-threats-what-is-social-engineering.html, accessed 19/02/20.

Kaspersky (n.d.). Social engineering – Definition. Available at: https://www.kaspersky.co.uk/resource-center/definitions/what-is-social-engineering, accessed 19/02/20.

### Blockchain Essay

The Internet Society www.internetsociety.org. (n.d.). Blockchain, available at: https://www.internetsociety.org/issues/blockchain, accessed 23/03/20.

Cointelegraph.com (n.d.). How Blockchain Technology Worlks. Guide for Beginners. Available at: https://cointelegraph.com/bitcoin-for-beginners/how-blockchain-technology-works-guide-for-beginners, accessed 23/03/20.

### Online Discussion Fora Essay

Johnson, C. S. (2017). Collaborative technologies, higher order thinking and self-sufficient learning: A case study of adult learners. *Research in Learning Technology.* 25. Available at: https://doi.org/10.25304/rlt.v25.1981, accessed 20/03/20.

Salmon, G. (n.d.). E-tivities – Introduction. *Gilly Salmon*. Available at: https://www.gillysalmon.com/e-tivities.html, accessed 20/03/20.

## Authors

**Clare S. Johnson**, **Dr. Ross Davies**, University of South Wales, J125, Pontypridd Campus, Pontypridd, Wales, CF37 1DL, Cardiff, United Kingdom, e-mail: clare.johnson@southwales.ac.uk