Developing a tool to add personalized watermarks to student work in order to prevent plagiarism

JULIUS KUIJF, University of Twente, The Netherlands

There are many forms of plagiarism which are indistinguishable from genuine work which causes problems for university-level institutions as students who get away with plagiarism are likely to do it again even after finishing their education. Genuine and plagiarized work is often differentiated from each other by comparing the work with work previously submitted, which is the method used by the primary plagiarism detection program used by universities known as Turnitin. This method of detecting plagiarism however cannot detect contract work, which is work done by a different person for some sort of compensation. In cases of contract work, the original is never published and thus not available for comparison. Other methods of detecting plagiarism however do exist, namely invisible watermarking. Invisible watermarking is a text-steganographic technique where invisible markers are added throughout a document so that the work can be traced back to its source. So far researchers have been able to differentiate between contract work and genuine work by looking at Open Office XML (OOXML) tags which are created during the authoring process by programs like Word. OOXML tags appear much more frequently in genuine work compared to contract work. These tags however have their limitations. It is not possible to see when they were made or by who, and they can easily be circumvented if a student is aware of their existence. This research attempts to expand onto this idea by developing a novel tool that adds invisible watermarks to student work intentionally whilst they are creating it. These watermarks will include their student id and a timestamp so identification is possible. These watermarks become invisible by encoding them into zero-width UTF-8 characters, and they are encrypted use a One-time pad so tampering becomes more difficult.

Additional Key Words and Phrases: Plagiarism, Text Steganography, Watermarking, Contract Work, Zero-width Characters, One-time Pad

1 INTRODUCTION

Plagiarism is a major issue for university-level institutions to where if left unchecked reduces credibility and incentives academics to plagiarise in the future[1]. Thus, it is essential to detect as many cases of plagiarism as possible. The most used plagiarism detection program, Turnitin, detects plagiarism by comparing the delivered work with all other works in its database and checking whether there is a significant overlap in the works [4]. This method is very effective for typical forms of plagiarism but falters at others, the most prominent of which is contract work. Contract work is work done by another person who has created the work genuinely, usually in return for some sort of payment. The technique used by Turnitin has no effect on this as the original text is not published online; It cannot be traced back to the original.

Previous research has instead used the steganographic technique known as invisible watermarking [2, 3]. Documents written in Word use Open Office XML (OOXML) to allow for certain authoring and rollback features. This feature of OOXML was exploited to discover differences between patch-worked text and text written genuinely. A large differences in the amount of these OOXML tags have been discover between the two works which indicates plagiarism. These tags are very fragile and do not contain very much information however. If a user copies their work and pastes it in a new document all the tags disappear. The tags also lack the kind of information that is desirable when determining if something should be considered plagiarism.

Another attempt at using invisible watermarking has been made using a different approach [6]. In this case, every student was given a personalized template with their student id embedded in it using invisible zero-width UTF-8 characters. A work was considered to be plagiarised if the student id of a different student showed up in a student's work. This research too had some limitations. Students had to enter their student id themselves which allowed for user error which actually caused some templates to indicate plagiarism even though the work was created genuinely. Also, if a student would be aware of these watermarks it would be very easy to replace it with a different one or simply remove it. Structured forms of text steganography are always weak to this form of attack compared to image steganography because text files have a lot less redundant data making the watermarks far easier to detect and separate from the rest of the file [5]. Both of these issues can be resolved by encrypting the watermark using a One-Time Pad [?], and by having the student authenticate themselves through the API of the relevant institution. The One-time pad makes it significantly harder to replicate a watermark for malicious purposes, and the API ensures that whatever student id is used within the watermark is the appropriate one.

In order to expand on the capabilities of plagiarism detection software especially in cases of contract work the goal is to develop a browser extension that a student has to log into using their university identification. Whenever the student makes an edit to their schoolwork the extension will create an authoring mark and add it to the document. The mark's information will include the student's identification and a timestamp. The timestamp is included to provide additional information so a more useful visualization can be produced. When a student submits their work the university will be able to decrypt these watermarks and tell how the document was composed.

Ideally, this browser extension would be able to apply these watermarks to all kinds of documents, such as source code, spreadsheets and images. However, that would take far too long for the allotted time. Instead, the extension will only apply to Google Documents. In future it would be possible to extend this extension further to other applications. Note that when Google Documents are exported to other formats (such as PDF) the watermarks will linger. For security reasons, it would also be better for the extension to connect to

TScIT 41, July 5, 2024, Enschede, The Netherlands

the API of the relevant institution. Chrome extensions are easy to tamper with so having these marks be generated by the extension itself is a security risk. New research could be done on this in the future.

2 RESEARCH QUESTIONS

This research attempts to expand onto the original OOXML method as well as the method used to add student id's to personalized templates with zero-width UTF-8 characters and integrating them into a watermark that is harder to tamper with [3, 6]. In order to know what is required for a tool to be successful at this task a few questions need to be answered.

2.1 RQ: What kind of tool is effective at preventing students from plagiarizing work using methods such as contract work and patch-working?

As stated before, unencrypted watermarks in text can be removed quite easily if the student is aware of its existence. If the tool is to be successful, students need to not be able to create their own watermarks. That is why it is necessary to log in to the extension using student credentials. Previous research shows that if students are allowed to type any student id in without verification then students often make mistakes giving incorrect information [2].

2.2 SRQ1: How can we differentiate between genuine work and plagiarized work?

When a student writes a text genuinely using Word or any other OOXML text editor they will leave authoring marks everywhere. A system that adds these authoring marks for the sole purpose of detecting plagiarism would be very effective at differentiating between the two works. The marks include the students identification number and a timestamp which gets encrypted and are then encoded into zero-width UTF-8 characters. Zero-width characters are invisible to readers when using regular text editors. Unlike OOXML tags these zero-width characters also get copied over when the text itself is copied. This allows students copying from other students to be detected as well. However, the primary way of detecting the difference between genuine and plagiarized work is not the absence of other student's marks, but the existence of the appropriate student's marks. If a student is aware of the existence of these watermarks it would actually be very easy to remove them which is not a problem.

2.3 SRQ2: How would a tool add these differentiators in a manner that is hard to tamper with?

The student is expected to log in using their university credentials. This is to avoid two problems: user error and intentional misrepresentation. If the student can simply enter any student's id they would be able to accidentally enter the wrong identification and cause trouble for both themselves and the student who the id actually belongs to. It would also allow students to create watermarks for other students, which would mean that contract work is still possible, although perhaps more limited. Since only the appropriate student is able to produce watermarks with their student identification encrypted within it we ensure that all watermarks are genuinely created. It is also worth noting that the student is not directly responsible for nor in direct control of when the watermarks get added. However, no actual connection will be made with the university API as it is outside the scope of this research. The tool will simulate a connection to the API but actually generate the data itself.

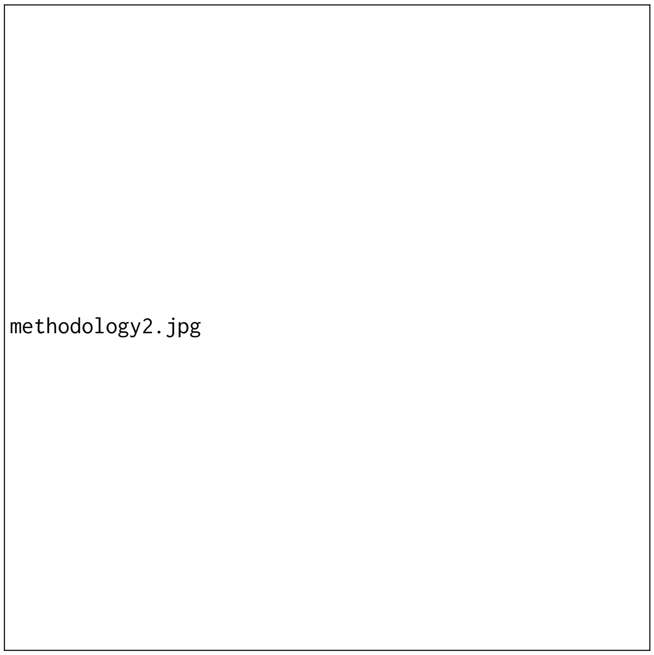

Fig. 1. Diagram of Methodology.

3 METHODOLOGY

The goal is to develop a tool that allows a university to differentiate between genuine and plagiarized work. In order to achieve that goal a few tasks have to be completed first as shown in Figure 1. Firstly, the actual tool that will add these watermarks needs to be developed. Then, samples need to be created which simulate the behaviour of a student creating genuine work and a student plagiarizing work in various forms, most prominently contract work. After that a visualization of the differences between the genuine work and plagiarized works is required to determine if a significant difference between the two works exist. Lastly, a conclusion can be drawn based on the visualization on whether or not the tool is effective at differentiating genuine and plagiarized work. The will be written in JavaScript as it is the most well-supported language for writing chrome extensions.

3.1 Develop the tool to add student-specific watermarks

Most of the work of the research will be allocated towards developing the chrome extension which will add the student-specific watermarks. Developing the tool consists of many smaller tasks. Here is a rundown of what the tool is required to do and how this is achieved.

(1) **Encrypt the watermark using a One-time pad** Encryption is not the focus of this research so a default implementation

provided by OneTimePad.js will be used to encrypt the message. The institution is the only party who is required to be able to encrypt and decrypt the message so a One-time pad (OTP) can be used. One-time pad is often unreliable because a key needs to be shared beforehand but because only one party is involved this part of the encryption process the key sharing step is not required and thus OTP is secure enough for the purposes of this research.

(2) **Encode the watermark into zero-width UTF-8 characters.** A method described in previous research will be used to encode the encrypted message. The same method as in previous research [6] will be used as this kind of encoding is resilient against being moved into different text formats. UTF-8 is widely supported. Student id's are encoded into bits, and then substituted by zero-width spacers and zero-width joiners for 0 and 1 respectively. A minor downside of this encrypting the information beforehand is that the resulting watermark is likely much longer than it is in the previous research. This has no effect on the robustness of the watermark however so it is only a minor inconvenience.

(3) **Avoid unintentional interaction with the watermark.** As a natural byproduct of the watermarks being encoded into UTF-8 characters, if a student puts their cursor in front of the watermark and then tries to remove text they will instead be removing part of the watermark and it would seem as if nothing was happening to the student. Previous research avoided this issue by putting the watermarks in specific spots where student would be less likely to interact with them[2]. This will not work for this tool as the location of the watermarks are out of our control. Instead this issue is solved by letting the tool move the cursor over the watermark to the appropriate side whenever a student happens to perform an action that would affect the watermark.

(4) **Allow reversal of encoded and encrypted watermarks** Decoding is done by returning the zero-width characters to their original bit representation. Decrypting is done by the library described as above so it is not a concern of the research.

(5) **Integrate placeholder watermarks within Google Documents.** In order to determine when watermarks need to be added to the document the tool needs to be able to detect when an edit is made. The tool will add a placeholder watermark that consists of just one zero-width joiner. This placeholder watermark procedure is not part of the final tool.

(6) **Integrate the actual watermark.** The dummy watermark can simply be replaced with the actual encrypted and encoded watermark.

3.2 Fabricate genuine and plagiarized work

Samples are required to determine the efficacy of the tool. There are a few different samples that need to be checked: genuine work, patch-working with watermarks, patch-working without watermarks, contract work, and replicated work. Genuine work will be created by taking a topic and writing a two paragraph essay about it. Patch-working with watermarks will be created by taking the genuine work and copying sections of it to a new document, then adding and removing a few sentences. Patch-working without watermarks will remove all the watermarks of the original work before making the edits. Contract work will take the genuine work and remove the watermarks but then make no additional edits. Lastly, replicated work will be created by reading the original work and retyping the entire essay. These sample types have been chosen as the tool has been designed only to catch these types of plagiarism. Example versions of these samples can be seen in [1] as well as a more detailed explanation of the differences between these works.

3.3 Visualize differences between genuine and plagiarized work

Both the numbers of watermarks, their timeline and their student identification number are important in distinguishing between genuine and plagiarized works. In order to visualize the data both a bar chart and timeline chart can be used. The bar chart will simply have the number of edits made for each document whilst the timeline chart is more complicated. What is expected of genuine work is for the student to make an edit periodically over the course of an hour or more, whilst plagiarized work is often done instantly or at least far faster than is reasonable. In addition, the percentage of watermarks per word will be supplied to account for the fact that longer text will have more watermarks on average. Typically visualizations of the detectability of the watermarks would be added as well. However, structured text steganography is always quite easy to detect if a person is aware of their existence [5] and students being able to detect and remove watermarks is not a concern for reasons already stated so these visualizations are not necessary.

3.4 Draw conclusions based on the visualization

If a significant difference can be found between the genuine work and the plagiarized works then this method of distinguishing between them will be successful. The differences between the works should be clear without doing any analysis as small differences between works would not be sufficient evidence in a realistic setting. The typical method of determining if something is plagiarism is by setting some form of threshold for the maximum allowed amount of copying and checking if the delivered work exceeds that threshold. If the threshold is exceeded then the work is considered plagiarized. Setting a threshold is controversial however [7], and because we cannot compare the plagiarized work to the original in the case of contract work an overlap threshold would have no effect. The differences between genuine work and plagiarized work is quite apparent so no threshold is required to determine plagiarism.

4 CONCLUSIONS

I believe that this tool will be very effective at combating the specific types of plagiarism outlined in this proposal. Although other methods have been used to differentiate between genuine and plagiarized work using steganography they have been quite easy to circumvent when the student is aware of their existence and often does not give enough information to come to a conclusion. That is why making a dedicated tool for creating watermarks that are more resilient

[1] https://docs.google.com/document/d/1-XbmcNJ3qkn6B1sYauiVaMoWQpaBuqTz9mZljwEPaFM/edit?usp=

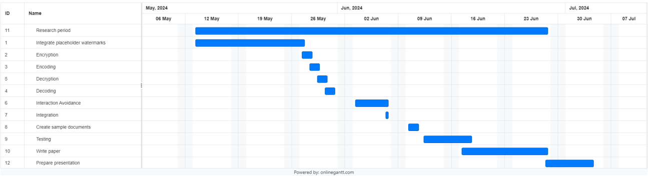

Fig. 2. Gantt chart for the research period.

against tampering and give the person looking for plagiarism more information a worthwhile investment.

4.1 Future work

It would be interesting to see if complete integration with the University of Twente API and Canvas can be done in future work. The visualization would be available to a trusted authority who is then expected to determine whether a work is suspicious based on the visualization. The work would also be authenticated using the actual API instead of a simulation, and the watermarks would be encrypted by a more secure algorithm than One-time pad.

REFERENCES

[1] Jorge Ávila de Lima, Áurea Sousa, Angélica Medeiros, Beatriz Misturada, and Cátia Novo. 2021. Understanding undergraduate plagiarism in the context of students' academic experience. *Journal of Academic Ethics* (2021), 1–22.

[2] Clare Johnson and Ross Davies. 2020. Using digital forensic techniques to identify contract cheating: a case study. *Journal of Academic Ethics* 18, 2 (2020), 105–113.

[3] Clare S Johnson and Ross Davies. 2020. Plagiarism from a Digital Forensics perspective. In *6th International Conference PAEB 2020 First Virtual ENAI Conference Conference Proceedings*. 37.

[4] Thomas Lancaster and Robert Clarke. 2014. Using Turnitin as a tool for attribution in cases of contract cheating. *The Higher Education Academy STEM* (2014), 1–5.

[5] Mohammed Abdul Majeed, Rossilawati Sulaiman, Zarina Shukur, and Mohammad Kamrul Hasan. 2021. A review on text steganography techniques. *Mathematics* 9, 21 (2021), 2829.

[6] Dylan Ryman, P.K. Imbrie, and Jeff Kastner. 2022. Enhancement of Plagiarism Detection Techniques via Watermarking. In *2022 IEEE Frontiers in Education Conference (FIE)*. 1–5. https://doi.org/10.1109/FIE56618.2022.9962396

[7] Steven L Shafer. 2016. Plagiarism is ubiquitous. , 1776–1780 pages.