2016

Detecting visual spoofing using classical cryptanalysis methods in plagiarism detection systems

Yang-Wai Chow
University of Wollongong, caseyc@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Ilung Pranata
University of Newcastle

Ari Moesriami Barmawi
Institut Teknologi Telkom

Detecting visual spoofing using classical cryptanalysis methods in plagiarism detection systems

Detecting Visual Spoofing using Classical Cryptanalysis Methods in Plagiarism Detection Systems

Yang-Wai Chow, Willy Susilo, Ilung Pranata and Ari Moesriami Barmawi

Abstract Plagiarism is a long standing problem that has been around for many years and remains a serious problem in key areas like education. Over the years, many plagiarism detection systems have been developed to automate the detection of plagiarised text in documents. However, researchers have shown that it is possible to employ visual spoofing techniques to defeat or cheat plagiarism detection systems. Visual spoofing refers to techniques used to alter information in a way that maintains a similar visual appearance as the original unaltered information. Using visual spoofing techniques, internal changes to the information stored in a text document can be done in a way that plagiarised text can escape detection by a plagiarism detection system's checking mechanisms. In addition, to a human reader the document will look perfectly legitimate and unsuspicious. This paper investigates the use of classical cryptanalysis methods to facilitate the identification of visually spoofed documents in automated plagiarism detection systems.

Yang-Wai Chow
Centre of Computer and Information Security Research, School of Computing and Information Technology, University of Wollongong, Wollongong, Australia, e-mail: caseyc@uow.edu.au

Willy Susilo
Centre of Computer and Information Security Research, School of Computing and Information Technology, University of Wollongong, Wollongong, Australia, e-mail: wsusilo@uow.edu.au

Ilung Pranata
School of Design, Communication and Information Technology, University of Newcastle, Callaghan, Australia, e-mail: Ilung.Pranata@newcastle.edu.au

Ari Moesriami Barmawi
School of Computing, Telkom University, Bandung, Indonesia, e-mail: mbarmawi@melsa.net.id

1 Introduction

Plagiarism has been defined by Potthast et al. [16] as the reuse of someone else's work while pretending it to be one's own. While there are many forms of plagiarism, such as the plagiarism of ideas, art, music, source code, etc., this paper focuses on the plagiarism of text in documents. Textual plagiarism has been a long standing problem that has been around for centuries and remains a serious problem in the key area of education [6]. This stems from the fact that plagiarism is inconsistent with pedagogical aims, as the mere copying of text has no conceivable educational value and is unfair to honest students [9]. Many researchers have suggested that the amount of plagiarism in this day and age has significantly increased and has been made easier due to the advancement of information technology, especially the Internet which makes access to information readily available [8, 10, 12, 13].

To address this serious problem, there has been much effort by researchers and practitioners in proposing and developing methods for automating the detection of plagiarised text in documents. To date, there are various commercial and non-commercial Plagiarism Detection Systems (PDSs) that have been developed over the years [16]. Many of these systems employ different approaches and strategies for detecting plagiarised text. In addition, the evaluation of automated PDSs based on a variety of criteria has been the subject of much research [4, 9, 12, 21, 22, 23]. These evaluation studies have shown that different PDSs are able to detect plagiarised text with varying degrees of success.

Several researchers have also highlighted a few simple tricks that people have employed to successfully defeat or cheat PDSs. The aim of these approaches is to disguise, alter or substitute the information in a text file, without much (if any) visible change to the visual appearance of text when the document is printed or displayed on screen [3, 6, 9]. The result of this is that to a human reader the text appears to be perfectly legitimate, readable and not in any way suspicious. However, regardless of the amount of plagiarised text contained in the modified document, it can pass through a PDS's checking mechanisms and the plagiarised text will escape undetected. It should be noted that while similarity detection is only one way of detecting plagiarism, it is one of the most fundamental approaches.

The disguising, altering or substitution of information, while attempting to maintain a similar visual appearance as the original information, is commonly referred to in the computer security community as visual spoofing. In the computer security domain, visual spoofing techniques have been deployed in a variety of attacks like Unicode and Internationalized Domain Name (IDN) homograph attacks [17], visual spoofing of Secure Sockets Layer (SSL) protected web sites [1], mobile application spoofing [11], as well as in various phishing scams [15]. These security attacks are designed to fool a user into believing that they are performing a secure transaction with a trusted service, when they are in fact disclosing their secret information to an attacker. This has led to various techniques being proposed to address visual spoofing in computer security [1, 11, 15, 17].

Our Contribution

This paper investigates the use of classical cryptanalysis methods to facilitate the identification of visually spoofed documents in automated PDSs. Cryptanalysis refers to the process of decrypting a message without knowing the underlying secret key [7]. The index of coincidence, frequency analysis and the χ^2 (Chi-Square) test are basic techniques that have successfully been applied in cryptanalysis, for example, for analysing and decrypting substitution ciphers. In this paper, we describe how these methods can be used to identify text documents that have been altered using visual spoofing techniques. We demonstrate the use of the proposed approach and present evaluation results on its effectiveness.

2 Background

This section presents a background to key areas of relevance to the research addressed in this paper. First, a description of visual spoofing methods that have been shown to be able to avoid detection in automated plagiarism detection systems is presented. This is followed by a description of three classical methods that have been used in cryptanalysis, namely, the index of coincidence, frequency analysis and the χ^2 test.

2.1 Visual Spoofing

Visual spoofing is a term that is commonly used in computer security to refer to the disguising, altering or substitution of information, while attempting to maintain a similar visual appearance as the original information. The aim of visual spoofing attacks is to trick users into believing that they are using the proper legitimate service, when in truth they are using a fraudulent service. While visual spoofing has been used for a variety of computer security attacks [1, 11, 15, 17], this paper addresses the problem of visual spoofing in text documents for the purpose of defeating or cheating automated PDSs.

Plagiarism detection systems work by first extracting text from a document. The extracted text is subsequently compared with other text sources, from either a customised database or from the Internet, or both [6]. Visual spoofing methods can be used to attack this by preventing the correct extraction of text from a document, to deter a PDS from identifying any plagiarised text contained within the document [6]. This must be done in a manner whereby the visual appearance of the document does not visibly change by much (if at all), to prevent a human reader from suspecting that the document has been disguised to escape detection when passed through a PDS.

A number of simple visual spoofing techniques have been identified by researchers and have been shown to be able to fool PDSs [3, 6, 9, 13]. Kakkonen and Mozgovoy [9] refer to these as technical tricks exploiting weaknesses of current automatic PDSs. Some of these are described as follows.

2.1.1 Homoglyph Substitution

The purpose of this approach is to substitute characters within a document with visually similar characters, but with different internal representations [9, 13]. This is possible using homoglyphs in Unicode. The Unicode Standard [20] is a character coding system designed to support the worldwide interchange, processing, and display of the written texts, and has the potential capacity of more than one million characters [17]. A glyph is an image that represents a character or part of a character. Hence, a homoglyph is two or more glyphs that cannot be differentiated by instant visual inspection [17]. There are many visually similar characters in Unicode. For example, the visual appearance of the letter 'O' when represented with three different Unicode characters looks similar: Unicode 004F (Latin O), 039F (Greek Omicron), and 041E (Cyrillic O) [9, 13]. By substituting certain characters in the text, some plagiarism detection algorithms like string-matching approaches, will not be able to match normal characters with their homoglyph counterparts.

2.1.2 Invisible Letters

Many text processors allow users to change the colour of characters. In this approach, extraneous letters are inserted into a document's blank spaces using white-coloured font (or a colour which is similar to the document's background) [9, 13]. To the naked eye, these invisible white-coloured letters would simply appear as blank spaces in the text. Thus, the document would be visually similar to the original, except for possibly some variation in the apparent blank spacing between words. Plagiarism detection software that extract the text will also extract these extraneous letters and treat them as part of the intended text. Hence, the inclusion of these extra letters in the text can successfully fool plagiarism detection algorithms.

2.1.3 Using Images

This technique exploits the fact that existing PDSs are incapable of comparing images [9, 13]. Plagiarised text can avoid detection when scanned text pages are inserted into a document as images. In this manner, the text completely bypasses the plagiarism detection mechanism, as it is not treated as text in the first place.

A related technique involves converting text to Bézier curves [6]. This method takes each character in the text and replaces it with a Bézier curve that defines the character's glyph, effectively replacing letters with visually similar images, which

will form the text that a human reader sees. Additionally, unrelated replacement text to fool a PDS can be inserted using the invisible text approach previously described in Sec. §2.1.2. This way, the text that the human sees is different from the text extracted by the plagiarism detection software. However, it has been highlighted that the use of images will greatly inflate the file size, possibly raising a reader's suspicion [6].

2.1.4 Changing the Character Map or Glyphs

Heather [6] describes a method of changing entries in a document's character map so that the visible text that a human sees in the resulting document will be different from the underlying text that a plagiarism detection software will extract. For example, one can shift the character map so that all displayed characters will be shifted by one character map position. In other words, whenever an 'a' appears in the document, a 'b' will be displayed instead; whenever a 'b' appears, a 'c' will be displayed; and so on [6].

Similarly, the glyphs (images that represent characters) can also be altered or rearranged [6]. This is possible because plagiarism detection software extract text before its translation into glyphs; whereas it is the glyphs that are displayed or printed, and is what a human reader sees. For example, if one were to swap the glyphs of 'x' and 'e', whenever an 'x' appears in the text the glyph showing 'e' will be displayed instead. Therefore, to trick a PDS all the instances of 'x' and 'e' in the text should be swapped. By doing so, the text that is extracted by a plagiarism detection software will have all occurrences of 'x' and 'e' swapped, but the glyphs that are displayed or printed will read correctly. Hence, by altering the character map or glyphs, what a plagiarism detection software extracts will be garbled text [6].

Plagiarism Detection Systems

It has been reported that the detection accuracy of plagiarism detection methods, like string-matching methods, which rely on character based similarity between documents decreases with increasing disguise of plagiarism [12]. Moreover, string-matching methods are the most commonly used methods adopted by automated PDSs to compare textual content between documents [9]. Studies have shown that the use of these technical tricks and disguises can potentially fool PDSs and that some major systems have yet to address these problems [3, 9, 12, 23].

It should be noted that some PDSs have taken steps to plug certain loopholes. For example, Turnitin, which is a popular PDS, states that the homoglyph substitution and invisible letter approaches can no longer be used to trick their system [19]. In the case of homoglyph substitution, the developers of Turnitin state that words that contain a special character will be matched against words containing every character that looks like that character; for invisible letters, Turnitin will not accept papers that appear to have this condition based on abnormal word lengths [19]. Nevertheless,

not all PDSs are able to handle these loopholes. A test conducted on 15 automated PDSs by Weber-Wulff et al. [23] found that *only* Turnitin was able to detect plagiarism in test cases involving homoglyph substitution, while another plagiarism detection software, Urkund, reported the use of non-Latin letters but did not find the plagiarised source.

2.2 Classical Cryptanalysis Methods

Cryptanalysis has been defined as the process of decrypting a message without knowing the underlying secret key [7]. More generally, it is the study of analysing cryptographic systems to discover exploitable weaknesses or vulnerabilities in such systems. The index of coincidence, frequency analysis and the χ^2 test are basic techniques that have successfully been applied in cryptanalysis, such as in breaking the Vigenère Cipher [7]. Furthermore, these methods have also been used in natural-language analysis, for example, to identify the language a piece of text was written in. A background of these methods is presented here, and in Sec. §3 we describe how these methods can be applied to the problem of identifying visual spoofing in text documents.

2.2.1 Index of Coincidence

The index of coincidence is a method that was invented by Friedman [2], who described its applications in the field of cryptanalysis. It also has uses in the field of natural-language analysis and can be defined as follows:

Definition 1. Consider a string, **s**, which consists of n alphabetic characters: $\mathbf{s} = c_1c_2c_3...c_n$. Let **IoC** denote the *index of coincidence*. The index of coincidence of **s**, **IoC(s)**, is the probability that two randomly chosen characters in **s** are identical [7]. It can be calculated as:

$$\mathbf{IoC(s)} = \frac{1}{n(n-1)} \sum_{i=0}^{k} F_i(F_i - 1) \tag{1}$$

Where n is the number of characters in **s**, and i represents the characters. Hence, for English, $i = 0, 1, 2, ..., 25$, representing the letters $a, ..., z$ (ignoring case sensitivity). F_i is the frequency with which letter i appears in **s** [7].

Based on the relative frequencies in which letters generally occur in a given language, an approximate Index of Coincidence (IoC) can be derived to identify whether or not a string of text is written in a particular language. Since letters will occur at different frequencies in different languages, each language will potentially have a different IoC value.

2.2.2 Frequency Analysis

Frequency analysis refers to the statistical analysis of the frequencies with which letters, or collections of letters, appear in a given text. In cryptanalysis, it is used to examine the frequencies of individual letters and their combinations in ciphertext to potentially identify exploitable weaknesses in a cipher [7]. Each language has a particular statistical distribution that can be used to characterise the occurrence of letters in a piece of text written in that specific language. For example, the characteristic distribution of letters in the English language is such that the letter 'e' has the highest frequency of occurrence, followed by the letters 't', 'a', 'o', 'i', 'n', etc., whereas the letters 'x', 'j', 'q' and 'z' occur at the lowest frequencies [14].

In addition, *n*-grams can be used to analyse combinations of *n* letters. Unigrams (*1*-grams) are individual letters, whereas bigrams (*2*-grams) and trigrams (*3*-grams) refer to combinations of two and three letters respectively. As an example, in the English language the bigrams 'th', 'he', 'in', have the highest frequencies of occurrence [14]. Depending on the text corpus that is sampled, the frequencies of occurrence can vary slightly. The work in this paper is based on the English language *n*-gram frequencies as reported by Norvig [14], which is based on the Google books *n*-gram dataset [5].

Based on the frequency of occurrence for individual letters, the IoC for the English language is ≈ 0.066 (calculated from the frequencies reported by Norvig [14]). If a string of text contains uniformly random letters, the probability that each letter occurs is $\frac{1}{26}$. Therefore, the IoC for random letters is expected to be $\approx \frac{1}{26} \approx 0.0385$. This means that if one were to calculate the IoC of a string of text, if the resulting IoC value is within range of 0.066, it is highly likely that the text is written in English, whereas if it is closer to 0.0385 it is likely to consist of random letters. The IoC can also be applied to bigrams, in which case $i = 0, 1, 2, ..., 675$ in Equation 1, representing the bigrams $aa, ..., zz$. For English language bigrams, the IoC is ≈ 0.00929 (calculated from the frequencies reported by Norvig [14]), whereas the IoC for uniformly random pairs of letters is $\approx \frac{1}{26*26} \approx 0.00148$.

2.2.3 The χ^2 Test

The χ^2 (or Chi-Square) test is a common statistical test that has many uses. It can be employed in cryptanalysis to compare potentially decrypted ciphertext candidates with the characteristic language distribution. The lower the resulting χ^2 value, the more likely the ciphertext has been successfully decrypted. The χ^2 test can be applied by calculating [18]:

$$\chi^2 = \sum_{i=0}^{k} \frac{(v_i - Np_i)^2}{Np_i} \tag{2}$$

Where N is the total number of samples, and p_i is the probability distribution of the ith set element. Hence, Np_i is the expected number of occurrences of the ith set

element, while v_i is the observed number of occurrences of the ith set element. A low χ^2 value indicates that the observed distribution closely matches the expected distribution, whereas in contrast, a high value indicates that the observed distribution deviates from the expected distribution.

3 Detecting Visual Spoofing in Text Documents

This section describes the notion underlying the proposed approach of using the index of coincidence, frequency analysis and the χ^2 test to identify anomalies in the textual content of documents. Such documents should be flagged as being suspicious as they do not conform to the norms of a given language. Some issues that should be considered and limitations of the approach are also discussed.

3.1 Proposed Approach

The aim of the proposed visual spoofing detection approach is to be able to detect whether or not the text contained within a document matches the characteristics of a given language. If the characteristics of the text deviates significantly from the expected language norm, the document should be flagged as being suspicious.

To illustrate this, consider a document that has been written in a particular language and contains plagiarised text. Furthermore, this document has been visually spoofed using the homoglyph substitution approach in an attempt to fool a PDS's plagiarism detection mechanisms. Since certain alphabetic letters in the text have been substituted with their homoglyphs, by calculating the IoC of the remaining letters in the text, one can use the result to determine if the text conforms to the expected value for that specific language. Text in a document that has been altered using homoglyph substitution will not be in line with the expected IoC value. The reason for this is because the characters that have been replaced with homoglyphs will not be included in the IoC calculation, as they fall outside the set of acceptable alphabetic letters of that language. Note that other characters like punctuations and spaces are also ignored in IoC calculations.

This same principle applies when attempting to identify documents that have been disguised through the altering of its character map or glyphs. As previously described in Sec. §2.1.4, the text that is extracted from such documents will be garbled text. Therefore, this meaningless collection of letters will not conform to the expected characteristics of the language, because the frequencies of n-grams in the extracted text will deviate from the language norm.

For the case where invisible letters are inserted into the text, if random letters are inserted, the resulting IoC of the text will be distorted based on the additional random letters. It is conceivable that instead of inserting random letters, the invisible letters can be carefully selected in a way that will maintain the expected distribution

of letter occurrences, hence keeping the individual letter IoC within an acceptable range. However, it will be highly unlikely, if not impossible, for the carefully selected letters to be inserted into the text while at the same time maintaining the expected bigram (and, if required, trigram) frequencies. Similarly, in the character map or glyphs alteration approach, if one were to carefully only swap letters with similar frequency of occurrence values so that the IoC will be within the acceptable range, bigram analysis using the χ^2 test will still reveal abnormal characteristics. Hence, it can clearly be seen why the proposed approach adopts the IoC, frequency analysis and χ^2 methods.

An issue that should be highlighted is that since this approach is based on the statistical characteristics of a particular language, the length and total number of words in a document will potentially have an effect on the accuracy of detection. Short documents may not entirely match to the expected characteristic distribution. In light of this, the detection thresholds may have to be relaxed for such documents.

Another issue that has to be considered is if the document was authored in poorly written language. For example, one can probably expect slight deviations in the statistical characteristics of the text if a document was written in a language that is not the author's native language, and/or contains many spelling errors. If plagiarism occurs in such cases, the text will typically be inconsistent; sections written by the actual author may be badly written, whereas the plagiarised sections will be written properly. In that respect, there may only be slight deviations in the statistical characteristics and not significant deviations as in the case of visually spoofed text documents.

3.2 Limitations

It should be noted that the proposed visual spoofing detection approach is based solely on examining the textual information that is extracted from a document. As such, it is unable to detect visual spoofing when images, as described in Sec. §2.1.3, are used to disguise the content. Nevertheless, as previously mentioned, the use of images should potentially raise suspicion in itself due to large file sizes in the resulting document [6].

In addition, the effectiveness of this approach will be significantly impacted when presented with documents that contain more than one language. This is due to the fact that the IoC and frequency analysis methods employed in the proposed approach are based on the characteristic statistical distribution of single languages. Furthermore, short lengths of text contained in short documents may have large statistical variations and may deviate from the expected distribution. Another point to consider is that different language datasets will have slight variations in terms of the language characteristics.

4 Evaluation and Discussion

This section shows how the proposed approach can be used in practice and presents various test results demonstrating its effectiveness. For the tests, transcripts of the following five famous speeches of various lengths were used:

- Sample 1: Abraham Lincoln's "Fourth of July Address" in 1861 (∼6.2k words)
- Sample 2: Winston Churchill's "Their Finest Hour" speech in 1940 (∼4.3k words)
- Sample 3: Julia Gillard's "Misogyny Speech" in 2012 (∼2.2k words)
- Sample 4: Steve Jobs' "Stanford Commencement Speech" in 2005 (∼2.2k words)
- Sample 5: John F. Kennedy's "Inaugural Address" in 1961 (∼1.3k words)

The characteristics of the original unaltered samples are first presented, as they can be referred to as the basis for comparison with results of their visually spoofed versions which will be shown later on. The graph in Fig. 1 compares the unigram (individual letter) frequencies in the samples to the expected English language unigram frequencies. It can be seen from the figure that in general, the unigram frequencies in the samples unsurprisingly conform to the expected frequencies. Table 1 shows the number of unigrams and bigrams in the samples, along with their respective IoCs. One can see that the IoCs of the samples are within range of the English language unigram and bigram IoCs of ≈ 0.066 and ≈ 0.00929, respectively. Since the unigram and bigram IoCs for uniform random distribution are ≈ 0.0385 and ≈ 0.00148 (as described in Sec. §2.2.2), respectively, larger variation in the bigram IoC is to be expected as the ratio between the expected IoC and the uniform random distribution IoC is greater compared with the unigram IoC's ratio.

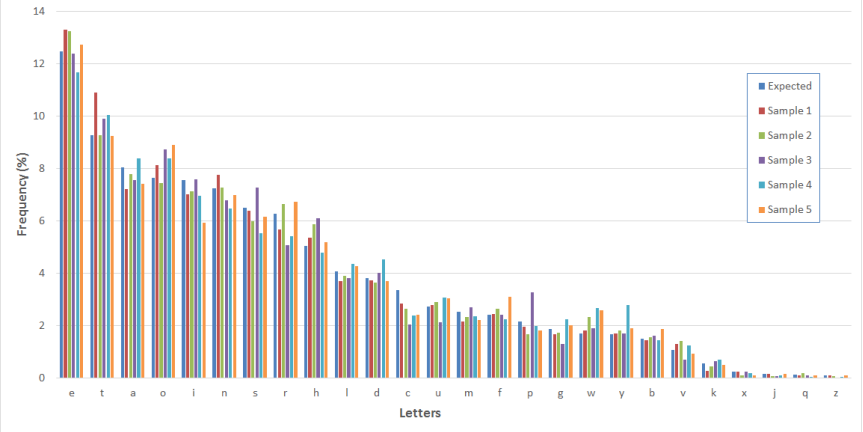

Fig. 1 Unigram frequencies in the samples compared with the expected English language unigram frequencies.

Table 1 Unigram and bigram characteristics of the samples.

	Unigram		Bigram	
Sample	**Number of Letters**	**Index of Coincidence**	**Number of Bigrams**	**Index of Coincidence**
1	29340	0.0695	23066	0.01111
2	19692	0.0671	15305	0.01039
3	10213	0.0678	7917	0.01181
4	9213	0.0643	6943	0.00937
5	5878	0.0658	4530	0.00969

To test the effectiveness of the IoC in identifying documents altered using homoglyph substitution, the Latin letters ‘o’, ‘c’, ‘p’, ‘y’, ‘a’ and ‘e’ in the samples were replaced with their Cyrillic or Greek homoglyphs. This is similar to the homoglyph substitution test conducted by Gillam et al. [3]. Table 2 shows the unigram IoCs of the altered samples. It can clearly be seen that the resulting unigram IoCs are very different from the expected English language IoC (i.e. ≈ 0.066). It is obvious that the bigram IoC will similarly also exhibit abnormal values.

Table 2 Homoglyph substitution results; unigram IoCs of text in the altered samples.

Sample	**Number of Letters**	**Index of Coincidence**
1	19028	0.0915
2	12881	0.0856
3	6569	0.0904
4	5933	0.0842
5	3809	0.0832

To evaluate the IoC’s ability to identify documents with invisible letters, the samples were modified using an approach similar to that used in Kakkonen and Mozgovoy [9]; the text in the samples was modified by replacing all the white blank spaces in the document with random white-coloured letters. The resulting unigram and bigram IoCs are presented in Table 3. One can see from the table that the unigram IoCs of the resulting samples are fairly different from the expected ≈ 0.066 value. To supplement this, results of the bigram IoCs are calculated and it can clearly be seen that these are very different from the expected value of ≈ 0.00929.

For the visual spoofing case involving the changing of a document’s character map, if one were to adopt a basic method akin to the simple example described in Heather [6] of shifting the character map by one position (i.e. ‘a’ in the document will be displayed as ‘b’; ‘b’ displays as ‘c’; and so on), it can easily be seen that the unigram distribution characteristics will be vastly different from the expected language distribution. This is illustrated in Fig. 2 for sample 1, where it can clearly be seen that the resulting relative unigram frequencies of the modified document deviate significantly from the language norm. The IoC value calculated from the modified document is 0.0746, which is very different from the expected value.

Table 3 Unigram and bigram IoC results, obtained after adding invisible letters to the samples.

	Unigram		Bigram	
Sample	**Number of Letters**	**Index of Coincidence**	**Number of Bigrams**	**Index of Coincidence**
1	34978	0.0605	34304	0.00633
2	23602	0.0585	23113	0.00583
3	12265	0.0588	11997	0.00649
4	11183	0.0560	10855	0.00509
5	7072	0.0574	6916	0.00545

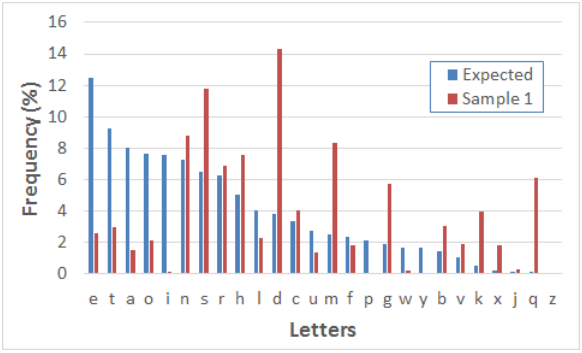

Fig. 2 Unigram frequencies in sample 1 as a result of shifting the character map by one position, compared with the expected English language unigram frequencies.

Instead of using a simple character map shift, an example of a better approach would be to swap letter glyphs between letters with similar frequencies of occurrence. For instance, the letters 'o' and 'i' have very similar frequencies of occurrence. Hence, if their glyphs were swapped and all instances of these letters were also swapped in the document (so that whenever an 'o' appears in the document, an 'i' is displayed instead; and vice versa), the IoC, which is based on the frequency of letter occurrences, would not be much different from the original document. However, by swapping letters in the document, the bigram frequencies will change. The χ^2 test can be used to identify anomalies in the resulting bigram frequencies in comparison with the expected frequencies. To test this, the following letters (which have very similar expected unigram frequencies) were swapped in the samples: 'o' and 'i', 's' and 'r', 'l' and 'd', 'm' and 'f', 'w' and 'y'.

The resulting χ^2 values of the samples are depicted in Table 4. The table shows χ^2 values (for all $p_i \neq 0$) obtain from the original and altered versions of the documents, as well as values calculated based on uniform distribution of bigram frequencies. It can be seen from the table that the altered version of the document has very high χ^2 values, and is comparable with values obtained if one were to assume uniform bigram distribution. Note that the χ^2 values calculated from the original documents are presented here for comparison purposes, because obviously

in practice only the visually spoofed version will be passed to a plagiarism detection system.

To explain the reason why the χ^2 values in the altered documents are so high, Table 5 presents the top 20 most frequently occurring English bigrams [14] and their expected numbers, together with the actual bigram counts from the original and altered versions of sample 1. The purpose of swapping the specific letters and their corresponding glyphs was to maintain a unigram distribution that would be similar to the expected distribution in the English language. Hence, the resulting IoC will not be much different from the expected value. However, it can be seen from Table 5 that the swapping of letters changes the bigram frequencies. Some of these changes are quite significant as highlighted by the boldface values in the table. This is the reason why the χ^2 values of the altered document are so high, because the bigram frequencies do not conform to the expected bigram frequencies of the English language.

Table 4 Bigram χ^2 values of the original document, the altered version and of uniform distribution.

Sample	Original Document	Altered Document	Uniform Distribution
1	4372.31	166342.04	150892.78
2	4119.42	137525.96	92836.76
3	4502.43	59636.70	55979.39
4	2471.32	67748.02	37638.05
5	1040.07	45441.74	25797.78

It can be seen from the results presented here that the proposed approach of using the IoC, frequency analysis and the χ^2 test, is effective in being able to detect anomalies in visually spoofed documents. The reason for this is because the text contained in an altered document will have a different unigram and bigram distribution compared with the expected distribution of a given language. As such, this demonstrates that the proposed approach is able to facilitate the detection of visually spoofed documents in automated PDSs.

5 Conclusion

This paper investigates how methods used for classical cryptanalysis can be applied to the problem of identifying documents that have been visually spoofed to trick automated plagiarism detection systems. In particular, this paper describes an approach of using the index of coincidence, frequency analysis and the χ^2 test for detecting anomalies in modified text documents. The effectiveness of the proposed approach was demonstrated on text documents that were modified using various visual spoofing techniques, namely, the homoglyph substitution, invisible letters and

Table 5 The top 20 most frequently occurring English bigrams and their expected bigram count, compared with the actual number of occurrences in the original and altered versions of sample 1.

		Bigram Count		
Expected Rank	**Bigram**	**Expected**	**Original Document**	**Altered Document**
1	th	821	1088	1088
2	he	708	834	834
3	in	560	470	541
4	er	473	464	319
5	an	473	402	402
6	re	426	406	**236**
7	on	405	541	470
8	at	343	390	390
9	en	334	391	391
10	nd	311	325	**27**
11	ti	309	319	250
12	es	309	319	464
13	or	295	308	284
14	te	277	326	326
15	of	270	296	**54**
16	ed	270	316	114
17	is	261	284	308
18	it	258	343	105
19	al	251	221	**90**
20	ar	247	200	207

changing the character map or glyphs approaches. The reason why the proposed approach is able to detect anomalies in visually spoofed documents, stems from the fact that the resulting distribution of unigrams and bigrams in the modified documents' textual content does not conform to the characteristic distribution of a given language.

References

1. A. Adelsbach, S. Gajek, and J. Schwenk. Visual spoofing of SSL protected web sites and effective countermeasures. *Lecture notes in computer science*, 3439:204, 2005.
2. W. F. Friedman. *The index of coincidence and its applications in cryptanalysis*. Aegean Park Press, 1987.
3. L. Gillam, J. Marinuzzi, and P. Ioannou. Turnitoff–defeating plagiarism detection systems. In *11th Annual Conference of the Subject Centre for Information and Computer Sciences*, volume 84, 2010.
4. T. Gollub, M. Potthast, A. Beyer, M. Busse, F. Rangel, P. Rosso, E. Stamatatos, and B. Stein. *Information Access Evaluation. Multilinguality, Multimodality, and Visualization: 4th International Conference of the CLEF Initiative, CLEF 2013, Valencia, Spain, September 23-26, 2013. Proceedings*, chapter Recent Trends in Digital Text Forensics and Its Evaluation, pages 282–302. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

5. Google. *Google Books Ngram Viewer*. http://storage.googleapis.com/books/ngrams/books/datasetsv2.html.
6. J. Heather. Turnitoff: Identifying and fixing a hole in current plagiarism detection software. *Assessment & Evaluation in Higher Education*, 35(6):647–660, 2010.
7. J. Hoffstein, J. Pipher, and J. Silverman. *An Introduction to Mathematical Cryptography*. Springer Publishing Company, Incorporated, 1 edition, 2008.
8. R. M. Howard. Understanding internet plagiarism. *Computers and Composition*, 24(1):3 – 15, 2007.
9. T. Kakkonen and M. Mozgovoy. Hermetic and web plagiarism detection systems for student essays An evaluation of the state-of-the-art. *Journal of Educational Computing Research*, 42(2):135–159, 2010.
10. A. Lathrop and K. E. Foss. *Student Cheating and Plagiarism to the Internet Era: A Wake-up Call*. Libraries Unlimited, Inc., Englewood, CO, USA, 2000.
11. L. Malisa, K. Kostiainen, and S. Capkun. Detecting mobile application spoofing attacks by leveraging user visual similarity perception. *IACR Cryptology ePrint Archive*, 2015:709, 2015.
12. N. Meuschke and B. Gipp. State-of-the-art in detecting academic plagiarism. *International Journal for Educational Integrity*, 9(1), 2013.
13. M. Mozgovoy, T. Kakkonen, and G. Cosma. Automatic student plagiarism detection: future perspectives. *Journal of Educational Computing Research*, 43(4):511–531, 2010.
14. P. Norvig. *English Letter Frequency Counts*. http://norvig.com/mayzner.html.
15. R. Oppliger and S. Gajek. Effective protection against phishing and web spoofing. In *Communications and Multimedia Security*, pages 32–41. Springer, 2005.
16. M. Potthast, T. Gollub, M. Hagen, J. Kiesel, M. Michel, A. Oberländer, M. Tippmann, A. Barrón-Cedeño, P. Gupta, P. Rosso, and B. Stein. Overview of the 4th international competition on plagiarism detection. In P. Forner, J. Karlgren, and C. Womser-Hacker, editors, *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*, volume 1178 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.
17. N. Roshanbin and J. Miller. Finding homoglyphs-a step towards detecting unicode-based visual spoofing attacks. In *Web Information System Engineering–WISE 2011*, pages 1–14. Springer, 2011.
18. B. Ryabko, V. Stognienko, and Y. Shokin. A new test for randomness and its application to some cryptographic problems. *Journal of Statistical Planning and Inference*, 123(2):365 – 376, 2004.
19. Turnitin. *Blog - Can Students "Trick" Turnitin?* http://turnitin.com/en_us/resources/blog/421-general/1650-can-students-trick-turnitin.
20. Unicode. *The Unicode Standard*. http://unicode.org/standard/standard.html.
21. J. D. Velsquez, Y. Covacevich, F. Molina, E. Marrese-Taylor, C. Rodrguez, and F. Bravo-Marquez. DOCODE 3.0 (DOcument COpy DEtector): A system for plagiarism detection by applying an information fusion process from multiple documental data sources. *Information Fusion*, 27:64 – 75, 2016.
22. D. Weber-Wulff. *Handbook of Academic Integrity*, chapter Plagiarism Detection Software: Promises, Pitfalls, and Practices, pages 1–11. Springer Singapore, Singapore, 2015.
23. D. Weber-Wulff, C. Möller, J. Touras, and E. Zincke. *Plagiarism Detection Software Test 2013*. http://plagiat.htw-berlin.de/software-en/test2013/report-2013/.