INTERNATIONAL JOURNAL OF ACADEMIC RESEARCH IN BUSINESS & SOCIAL SCIENCES

Vol 15, Issue 12, (2025) E-ISSN: 2222-6990

Protecting Digital Documents in E-Learning: A Novel Ownership Verification Mechanism Using Steganography and QR Codes

Nor Fatin Shazwani Adnan, Nurhashikin Mohd Salleh*, Siti Rahayu Selamat, Mohd Zaki Mas'ud

Fakulti Kecerdasan Buatan dan Keselamatan Siber, Universiti Teknikal Malaysia Melaka,
Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia
Corresponding Authors Email: nurhashikin@utem.edu.my

DOI Link: http://dx.doi.org/10.6007/IJARBSS/v15-i12/27025

Published Date: 18 December 2025

Abstract

COVID-19 has created the worst crisis for businesses worldwide, particularly in the education and learning industries. During the Covid-19 pandemic, most learning institutions will have to use the digital environment to communicate knowledge and skills to their learners. E-Learning Management System is used to share learning resources in form of digital documents for the teaching and learning process. However, digital documents are vulnerable to unlawful reproduction and distribution. Thus, it is necessary to safeguard and establish document ownership. Based on this reason, this paper proposes a mechanism to safeguard and confirm ownership of documents using the steganography technique embedded in QR code technology. The main objective of this study is to design and implement a mechanism that can securely embed ownership and acquisition information into digital documents. The scope of this study is limited to embedding and verifying ownership information; it does not address broader cybersecurity controls such as full document encryption or digital rights management. In the proposed mechanism, information related to the documents such as the owner and the person who acquired the document is concealed using the steganography technique and embedded in QR Code. A prototype is developed and tested based on the criteria of completeness, usability, and imperceptibility parameters to verify the effectiveness of the mechanism. The result indicates that the proposed mechanism can confirm the ownership and protect the documents from any misused activities such as being distributed on a public platform without the owner's consent.

Keywords: QR Code, Document Protection, Ownership, Steganography, E-Learning Management System

Introduction

Educators and learners currently use E-learning management systems as a platform for knowledge and skill sharing. According to Dhawan (2020), the COVID-19 pandemic has significantly transformed the educational sector, leading to a rapid transition towards online teaching and learning. This transition has revealed emerging patterns in educational methodologies, creating chances for people to pursue higher education or further their studies via adaptable and easily available online platforms. The widespread availability of Massive Open Online Courses (MOOCs), blended learning models, and asynchronous learning settings has made education more accessible to a larger number of people, allowing for continuous learning and career advancement on a previously unseen level. (Dhawan, 2020; Hodges et al., 2020; Bao, 2020; UNESCO, 2021). The transition to E-Learning platforms has brought about various challenges, particularly concerning the use of digital documents. One significant issue is copyright infringement, as highlighted by Dang et al. (2019). The pervasive sharing of digital content has made it easier to distribute copyrighted materials without proper authorization, leading to legal and ethical concerns. Additionally, the issue of document forgery is a significant concern that is closely linked to digital learning settings. As stated by Ahvanooey et al. (2018), altering digital documents might compromise the reliability and genuineness of educational records, which can have significant consequences for credibility of academic qualifications. Aru and Ananaba (2018) also stated the widespread availability of digital copies makes these issues worse because it is harder to control and monitor the spread of original content.

Therefore, the objective of this study is to safeguard and confirm ownership of digital documents in the education industry. As a result, QR code technology and a steganography approach have been used as an ownership protection mechanism to overcome the issue. The information about the owner's file and the person who acquired is stored and concealed in the QR Code via steganography technique. A watermark with the logo of the respective institution is added to provide further security to the digital documents. Hence, the proposed mechanism can trace the culprit based on the information discovered from the QR code. The rest of this paper is organised as follows. Section II is the related work of the previous literature review. Section III discusses methodology of this study. Section IV presents the results from the study, while Section V provides implementation and implications of the results, as well as the limitations of the study. Section VI concludes the paper and provides several suggestions for future research.

Related Work

Currently, the advancement of technology and communications enables individuals to access the internet more quickly and to receive information from the internet more efficiently. In addition, the distribution and sharing of digital documents are simple, whereas shared data ownership and copyright protection are essential. For instance, during the Covid-19 outbreak in 2020, the world depended on the internet to transmit data, share resources and communicate with each other. The outbreak also affects most countries' teaching and learning environments, especially Malaysia. Due to the physical restrictions, new strategies were immediately taken by all educational institutions in Malaysia (Jaafar et al., 2022), which include online teaching and learning. However, this strategy creates a new challenge in protecting ownership and copyrights of the resources. Numerous studies have been conducted to safeguard ownership and copyrights to overcome this issue.

One of the easiest methods to protect against these problems is using Quick Response (QR) code that only needs a QR code scanner. Hassan and Hussein (2020) stated that QR code is the fastest way and consumes less cost in transferring data. However, the data stored directly to the QR code is not secure. Thus, Ashwini (2021) proposed a standard multi-color QR code based on texture patterns and text steganography to hide data. The proposed solution implements a visual secret sharing scheme (VSS) before storing the data in the QR code. Consequently, the proposed solution can improve two aspects of their proposed idea: security and the partitioning technique. Huang et al. (2020) proposed using two watermarking types with the application of a QR code for image copyright protection. Their research applied the Discrete Cosine Transform (DCT) as the key component of image compression, and there are two watermarks embedded into the color images. The first watermark is the QR code containing the desired copyright information, and the second is a binary random sequence. As a result, the proposed method has successfully created a robust watermarked and improved copyright protection mechanism for documents.

In contrast, Arkah et al. (2019) proposed multiple QR code generation using a color map as a digital signature for a document. The digital signature generates from the color extracted in the document. The digital signature is generated using the retrieved colour from the document. The extracted colour will next undergo a process to become a colour map, serving as the document's digital signature. The QR code is then imprinted on the page after the digital signature has been put into the numerous QR codes. Consequently, the suggested approach can determine whether or not the document has been altered. Meanwhile, Li et al. (2017) have implemented the Discrete Wavelet Transform (DWT) and Singular Value decomposition using QR code technology to combat counterfeiting difficulties (SVD). In their research, the information about the copyright holder is rendered as an image. The image will then be placed into the QR code and transformed into a watermark to address the issue of counterfeiting. Therefore, this strategy can preserve copyright material despite being subject to various attacks.

Steganography is one technique used to protect ownership and copyrights of resources other than the QR code technology. According to Kadhim et al. (2018), the term "Steganography" is derived from the Greek words "Stegano" and "Graphy," whose combined meaning is "cover writing." There are various varieties of steganography, including audio, video, and image. Image steganography is hiding information such as an image, video, or text into another image that will cover the information (Subramanian et al., 2021). This steganography technique can protect the information from being tampered (Amarendra et al., 2019). Kumar et al. (2021) stated that higher capacity is necessary for image steganography technology for enhanced imperceptibility. In image steganography, data concealment techniques include least significant bits (LSB), masking, filtering, and transformation (Thampi, 2014). Due to its straightforward approach to concealing information in a cover image, LSB is the most popular method employed by researchers.

Manimekalai and Bakkiyalakshmi (2017) have used a steganography technique for information hiding. The researchers proposed a way to hide data in QR codes with the combined concepts of steganography and cryptography. The technique creates a secret message and embedded the message in a QR code. Then, the QR code with the hidden message is encrypted and embedded in a cover image using the least significant bits (LSB) insertion technique that creates a steganography image. Alajmi M. et al. (2020) introduced a steganography system that uses a QR code as a container to conceal the secret message (payload) and provide false information to the attacker. In their research, everyone with access to the system can read the message, but a secret key is required to obtain the payload. Using a QR code reader, the researchers evaluate the similarities between their QR codes and the standard QR code. The research discovered the generated QR code number of pixel change (NPCR) is less than the ordinary QR code, which indicates the better similarity of the images.

Methodology

In this study, there are six phases to be carried out. The phases are Literature Review, Analysis, Formulate hybrid copyright protection technique, Embedded protection techniques into document, Implementation, and Testing are shown below.

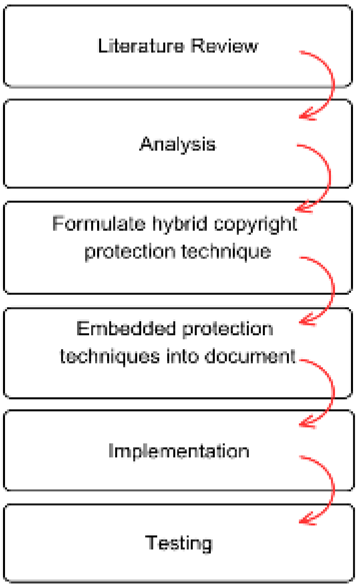

Literature Review
Analysis
Formulate hybrid copyright protection technique
Embedded protection techniques into document
Implementation
Testing

Fig. 1: Methodology

The description of the phases is explained below.
Phase 1: Literature Review
To conduct this phase, several sub-phases are involved. Initially, the topic was defined and the scope was outlined to include both technical and legal technique. Then, a systematic search was done across academic databases such as Google Scholar and IEEE using relevant keywords. After that, relevant studies were chosen according to predetermined criteria for inclusion and exclusion with a specific emphasis on those that directly addressing document protection in E-Learning. An in-depth study was conducted to assess the dependability and relevance of the findings.

Phase 2: Analysis
In this phase, an analysis of the document copyright protection technique is conducted. During this analysis, it is found that three techniques can be used for document protections which are Watermarking, Steganography, and Cryptography. The second analysis is the analysis of the Watermarking and Steganography technique with the QR code application. During the analysis, it is found that both Watermarking and Steganography techniques are suitable techniques to be used since their purpose is among the most related to this works.

Phase 3: Formulate hybrid protection technique
During this phase, the QR code generator process, and steganography image generator process has been designed. This includes which user interface of the prototype system that this module is going to be located, what is the required information that is going to be used in the module, how to generate the QR image and steganography image, where to saved the images, and how to pass the images to the other module.

Phase 4: Embedded protection techniques into document
This is another important phase after formulating the QR code generator and steganography image generator. To make a document with a QR code image, steganography image, and watermark image embedded within it, it first needs to be structured. There are some information that needed to be collected which are current user information who is downloading the document, for example, ID number, user name, and user email, user matrix number, a material selected to download by the current user, for example, lecture note or lab sheet, the material selected information and properties, for example, the lecture note and lab sheet owner name, its source, the desired width and size of the QR code image, steganography image and watermark image to embed into the selected material, position to locate the images, page to locate the images and where to store the embedded material.

Phase 5: Implementation
In this phase, the implementation of the system module is conducted. A PHP programming language will be used to write the code and all the system modules are created including integrating the QR code generator module, steganography image generator module and embedded the QR code image, watermark image, and steganography image into the document module with the system. During this process, it is important to make sure all the systems function and integrated function works well without a problem to avoid problem occur later in the testing part.

Phase 6: Testing
In this phase, each module of the system will be tested. The success of generating the QR code image, generating the steganography image, and embedding the generated QR code image, generated steganography image, and watermark image into the digital document is tested to ensure that it works as planned. Testing with several parameters is also performed.

Implementation and Result

Fig. 2 shows the diagram for the whole process and how it was connected to each other

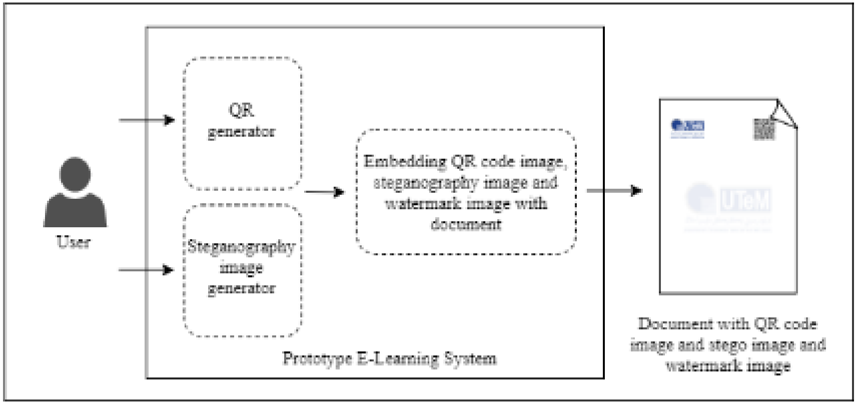

QR generator
Steganography image generator
User
Embedding QR code image, steganography image and watermark image with document
Prototype E-Learning System
UTeM
Document with QR code image and stego image and watermark image

Fig. 2: *System Architecture*

Based on Fig. 2, there are three main modules in this works which are the QR code generator, steganography image generator, and embedding the images into the document. The language used for implementing the prototype system is PHP language. Each module has its processes which are:

QR Code Generator
For QR code generation process, there are eight processes involved which shows in Fig. 3.

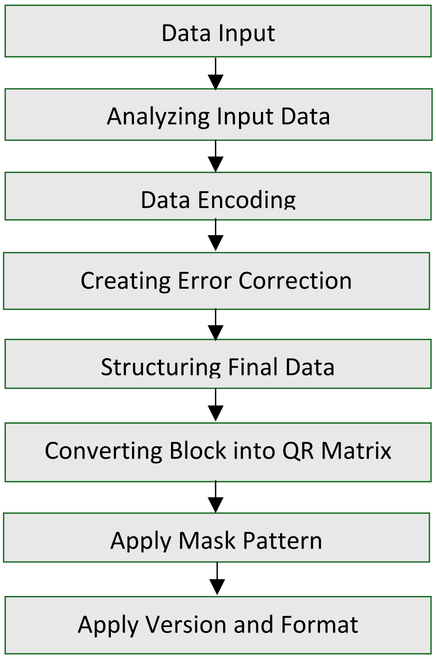


Fig. 3: QR Code Generation Process

Based on Fig. 3, the eight processes in generating the QR code image are data input, analyzing input data, data encoding, creating error correction codewords, structuring final data, converting block into QR matrix, apply mask pattern, apply version, and apply format information. The description for each process are:

Data Input
Data input is the information that has been gathered either from the prototype system or from the program. The informations are the user id, user name, user matrix number, user email, owner of the downloaded document, and source of the downloaded document. While the information gathered from the program is error correction capability (ECC) level, pixel size, and frame size.

Analyzing Input Data
During this process, the data input type from the data input process is being analyzed. The data input will be determined whether it is numeric, alphanumeric, byte, and kanji data type.

Data Encoding
After the data input type has been determined, the data input is encoded according to the data type encoding. The input data will be transformed into a string of bits according to the respective data type. Before that, the QR code version and maximum allowable capacity for the QR code will be determined. For example, a phrase 'NAME: FATIN' as input data. After

has been analyzed, the data type is found as alphanumeric and the ECC level being setup is level low. Table 1 will show version 1 as its smallest version which the capacity for the QR code can store the data is maximum 25 characters.

Table 1
QR code version with maximum allowable capacity (Denso wave Incorporated, 2021)

Version	Modules	ECC level	Numeric	Alphanumeric	Byte	Kanji
1	21x21	L	41	25	17	10
		M	34	20	14	8
		Q	27	16	11	7
		H	17	10	7	4

The next subprocess is determining the mode indicator. Each data type has its mode indicator. For instance, Numeric is indicated as 0001, Alphanumeric is 0010, Byte is 0100 and Kanji is 1000. After the indicator mode is determined, the next process that needs to be determined is the character count indicator. In this process, the characters counting will start from the original input data and then convert it into binary. The character count indicator's length is based on the encoding type and the QR code version used. For example, from the same phrase, 'NAME: FATIN' which consists of 11 characters with QR code version 1 and encoding data type Alphanumeric, then the character count indicator must have a length of 9 bits as shown in Table 2.

Table 2
Character count indicator according to the version and data type (Thonky, 2021)

Version	Data type	Character count indicator
1 - 9	Numeric	10 bits
	Alphanumeric	9 bits
	Byte	8 bits
	Kanji	8 bits

Based on Table 2, 11 characters that have been converted into a binary need to fits in the length of 9 bits. The 11 characters become 1011 in binary. To make it fits in the length of 9 bits, it will become 000001011 which the 0s will be filled on the left side if the binary is not enough 9 bits. This character count indicator result will be added after the mode indicator which will be 0010 000001011 [mode indicator] [character count indicator]. Table 3 shows the summary for the sub process explanation:

Table 3
Summary for sub process explanation

Data Input	NAME: FATIN
Data Type	Alphanumeric
ECC Level	Low
Version	1
Maximum to store data	25 characters
Total Characters	11
Characters in binary	1011
Mode Indicator	0010
Character count indicator	9 bits
Current Bit String	[Mode Indicator] [Character Count Indicator] 0010 000001011

Based on Table 3, the mode indicator and character count indicator has been determined. Next, the input data then will be encoded according to the selected data type encoding. Table 4 shows the encoding process for alphanumeric type.

Table 4
Alphanumeric Data Encoding

	Phrase in pairs					
Character	N A	M E	: SPACE	F A	T I	N
Code	23 10	22 14	44 36	15 10	29 18	23
Calculation	**Formula = (First character * 45) + second character**					
Calculation	(23*45) + 10	(22*45)+14	(44*45)+36	(15*45)+10	(29*45)+10	23
Decimal	1045	1004	2016	685	1315	23
Binary	10000010101	1111101100	11111100000	1010101101	10100100011	010111

Based on Table 4, each character will be separated into pairs, converted into decimal according to the alphanumeric table, multiplied the first character value with 45 and plus the second character. The calculations will result as a decimal number and need to be converted into binary. If the phrase or the input data in pairs is odd, the character will be converted into a 6-bit binary (in the example the odd character is the last N) as shown in Table 4. The current bit string after the phrase is encoded is shown in Table 5.

Table 5
Current bit string for the example

Mode Indicator	**Character count indicator**	**Encoded data**	**Total bit string**
0010	000001011	10000010101 1111101100 11111100000 1010101101 10100100011 010111	72 bits

Next, the current bit string will be broken up into 8-bit codewords and bytes will be added if necessary. In this sub-process, the required bit for the QR code is determined. Table 6 show some of the required data codewords for QR code version 1.

Table 6

Number of data codewords (Thonky, 2021)

Version – Error Correction (EC) Level	Total Number of Data Codewords for this Version and EC Level
1-Low	19 bits
1-Medium	16 bits
1-Quartile	13 bits
1-High	9 bits

As stated in Table 3, the ECC level is low, version is 1 and according to Table 6, the required total number of data codewords for QR code version 1 and ECC level low is 19 bits. Therefore, the total bits required for the QR code is 19 * 8 bits which is equal to 152 bits. Next, the indicator of 0s is added to the right side of the encoded data if the total bit string is shorter than the total of bits required in the QR code. For example, the current example bits string is 72 bits in Table 5, but the required bits for the QR code is 152 bits long. Because of that, the terminator is needed but the terminator can only be at most 4 bits long, so four 0s are added to the right side of the encoded data. Table 7 shows the terminator is added to the right of the encoded data and resulted in the current bit string to 76 bits.

Table 7

Added terminator value

Mode Indicator	Character count indicator	Encoded data	Terminator	Total bits string
0010	000001011	10000010101 1111101100 11111100000 1010101101 10100100011 010111	0000	**76 bits**

Next, the string of bits is arranged into 8-bit. According to Table 7, the string bits are not enough of 8-bits on the Terminator tab, then more 0s are added on the right side of the data bits. Table 8 shows the bits string of the encoded data that is arranged in 8-bit by 8-bit. The zero value is added on the right side to completed the 8-bit.

Table 8

Arranged encoded data

Bit string in 8-bit	Total bits string
00100000 01011100 00010101 11111011 00111111 00000101 01011011 01001000 11010111 00000000	**80 bits**

According to Table 8, the current bit string still does not reach the total required bits which is 152 bits. To achieve the 152 required bits for the QR code, a byte of 11101100 (236 in decimal) and 00010001 (17 in decimal) can be added repetitively until the total bits string achieve the required bits. To calculate how many bytes are needed to be added, the total required bit string can be minus with the current bits string, for example, 152 bits (required bits) – 80 bits (current bits) = 72 bits left. To convert in a byte, divided the result with 8 for example, 72 bits/8 = 9 bytes. Therefore 9 bytes are required and must be added to the end of the data string as shown in Table 9.

Table 9
Added required bytes

Required bytes	**Total bits string**
00100000 01011100 00010101 11111011 00111111 00000101 01011011 01001000 11010111 00000000 11101100 00010001 11101100 00010001 11101100 00010001 11101100 00010001 11101100	**152 bits**

After the data has been encoded and meets the required bits such as in Table 9, the next process is creating an error correction codewords for the data.

Creating Error Correction Codewords

In this process, the encoded data in the previous process will be converted back into decimal and polynomial forms. There are several calculations involved such as multiplying and XORing. After the data codewords and error correction codewords have been created, the data will be structured.

Structuring Final Data

In this process, the final data will be structured in a block. After the final data has been structured in a block, it will be placed into the QR matrix.

Converting Block into Qr Matrix

The data that has been structured in a block in the previous process will be placed into the QR matrix along with the finder pattern, timing pattern, alignment pattern, and separator.

Apply Mask Pattern

In this process, the data masking is applied to the data that have been placed in the matrix by toggle the module color. This is to ensure that the QR code is readable by the QR scanner.

Apply Version and Format Information

In this process, the version and the format information are applied to the QR matrix. Lastly, after the required information has been placed in the QR matrix, a quiet zone is applied around the QR matrix which will become a complete QR code.

Steganography Image Generator

For the steganography image generation process, there are five processes involved which are shown in Fig. 4.

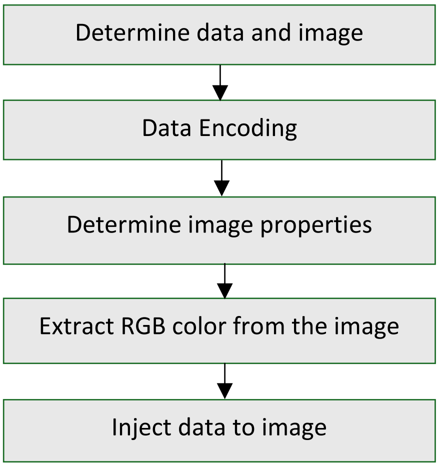


Fig. 4: Steganography image generation process

Based on Fig. 4, the five processes in generating the steganography image are determined data and image, data encoding, determine image properties, extract Red, Green, Blue (RGB) color from the image, and inject data to the image. The description for each process are:

Determine Data and Image

In this process, the data is being determined the same as the data to be encoded in the QR code image. The image that will be a container to store the data is determined by giving the path of the image itself. The image chosen in this works is the university‘s logo.

Data Encoding

The data that has been obtained in the previous process is encoded by converting each of the data characters into an integer based on the ASCII table which then will be converted again into binary and ‘end of text’ from the ASCII table will be inserted into the end of the encoded data.

Determine Image Properties

In this process, the properties of the image container are being determined to be manipulated. In this works, the image properties that have been used to manipulate are image opacity and image transparency.

Extract RGB Color from the Image

After the image container has been manipulated as suitable for this works, the RGB channel of the manipulated image will be extracted. The reason to extract this RGB value is that the encoded data will be injected into the least significant bit (LSB) of the blue channel. The blue channel is used because human eyes are less sensitive to that color. According to Vaishnavi and Subashini (2015), there are three channels where the eyes can sense more which are red (R), green (G), and blue (B) and they stated that 65% of the eyes are sensitive to red color, 33% are sensitive to green color and 2% are sensitive to blue color. Therefore, the blue channel is chosen to inject the encoded data as it is the lesser can be detected by eyes.

Inject Data to Image

After the RGB color from the image has been extracted, the value of the blue channel will be converted into binary form. Then, each binary number of the data, for example, character 'U', in binary is 01010101, this binary number will be injected into the LSB of the image's blue channel starting from the image's top-left corner and, process each image's pixel row until the entire message has been injected and produced a steganography image.

Embedding Images into Document

In embedding the images into the document process, there are four processes which are import document, determine page in the document to put the images, import images, and embed images into the document as shown in Fig. 5.

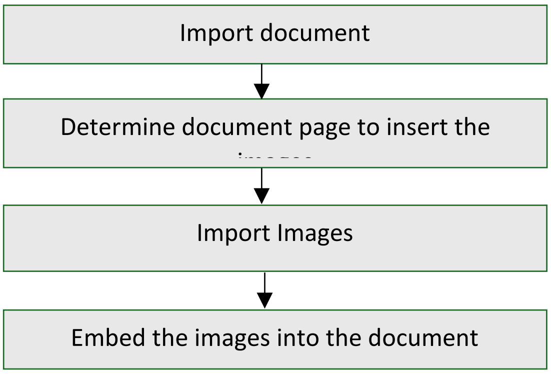

Import document
Determine document page to insert the
Import Images
Embed the images into the document

Fig. 5: Embed images into document process

Based on Fig. 5, the image that will be embedded into the document are the QR code image, steganography image, and university's logo as a watermark image into the document. The description for the processes are:

Import Document

In this process, the host of the three images are being imported. The document is imported by getting the document from the user who using the prototype system. Once the user from the prototype system clicks document from the system, that document will be the document that will be embedded with the images stated.

Determine Document Page to Insert the Images

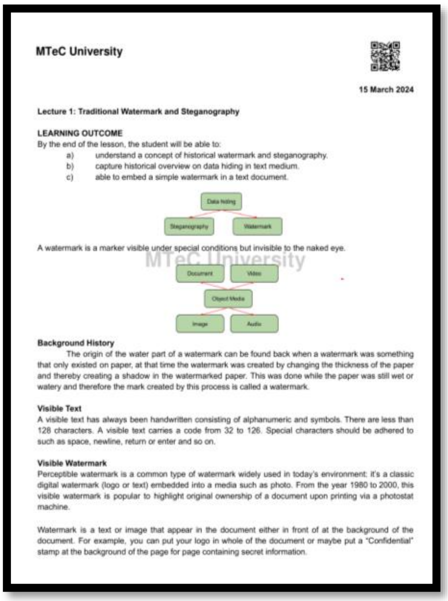
In this works, the page to embed the three images is decided to be all the document pages. A loop is made in the program to iterate through all pages.

Import Images

In this process, the images are being imported by giving the path on where the three images are stored in the local storage.

Embed the Images into Document

After the path of the three images has been defined, the three images will be embedded into the imported document. As a result, a pdf document with a QR code image, a steganography image, and a watermark image embedded within it is produced and is saves on the local server. Lastly, this document is the document that is going to be downloaded by the user in the prototype system. Fig. 6 shows the result of this works.



MTeC University

15 March 2024

Lecture 1: Traditional Watermark and Steganography

LEARNING OUTCOME
By the end of the lesson, the student will be able to:
a) understand a concept of historical watermark and steganography.
b) capture historical overview on data hiding in text medium.
c) able to embed a simple watermark in a text document.

A watermark is a marker visible under special conditions but invisible to the naked eye.

Background History
The origin of the water part of a watermark can be found back when a watermark was something that only existed on paper, at that time the watermark was created by changing the thickness of the paper and thereby creating a shadow in the watermarked paper. This was done while the paper was still wet or watery and therefore the mark created by this process is called a watermark.

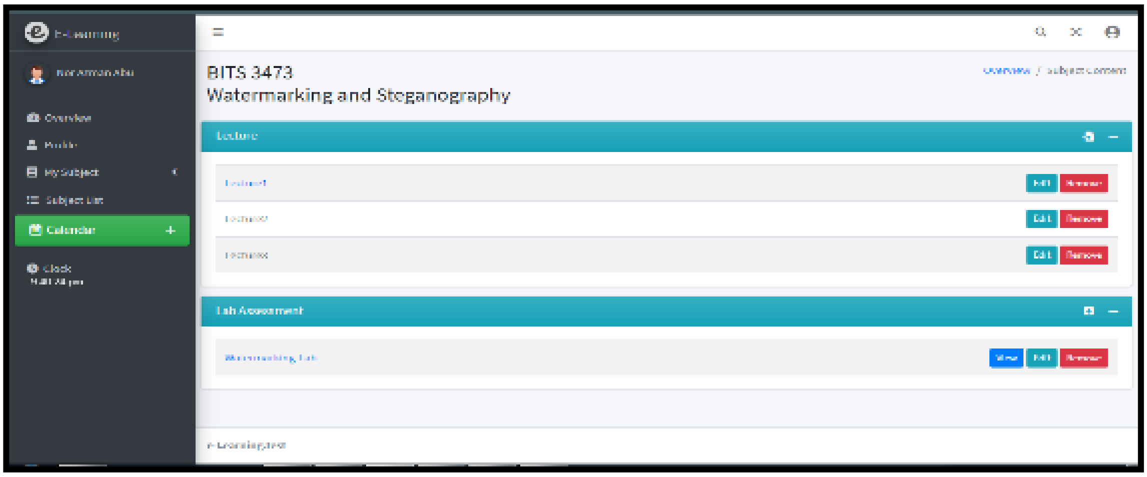
Visible Text
A visible text has always been handwritten consisting of alphanumeric and symbols. There are less than 128 characters. A visible text carries a code from 32 to 126. Special characters should be adhered to such as space, newline, return or enter and so on.

Visible Watermark
Perceptible watermark is a common type of watermark widely used in today's environment: It's a classic digital watermark (logo or text) embedded into a media such as photo. From the year 1980 to 2000, this visible watermark is popular to highlight original ownership of a document upon printing via a photostat machine.

Watermark is a text or image that appear in the document either in front of at the background of the document. For example, you can put your logo in whole of the document or maybe put a "Confidential" stamp at the background of the page for page containing secret information.

Fig. 6: Digital document with QR code image, steganography image, and watermark image embedded within it

Based on Fig. 6, positioning on the top-right is the QR code image, on the top-left is the university logo that acts as a watermark, and on the center is the steganography image. Several parameters have been chosen to measure the effectiveness of the protection technique. The embedding process is being tested by login into the prototype system and download any document from the prototype. Fig. shows one of the user interface to download the digital document.



BITS 3473
Watermarking and Steganography

Fig. 7: User interface to download digital document

Based on Fig. 7, the digital document that can be downloaded is the lecture note document. The successfulness of this testing can be measured by seeing the document display in a new tab with the three images embedded. For testing on the QR code image, there are two parameters involved which are completeness of data and usability. In QR code image testing, the information after scanning the QR code will be compared with the data from the system and database. The analysis of the data that has been compared is shown in Table 10.

Table 10
QR code testing analysis on data completeness

Properties	Data comparison	Advantages	Disadvantages
Accurate	✓	The data is accurate	The data shows after scanning might be too much.
Complete	✓	The data is complete	

Based on the analysis in Table 10, it can be concluded that the information stated in the QR code is accurate and complete but the ownership information and the downloading activities information from the QR code might be too much. While for usability parameter, user usability testing on the QR code image is conducted and the analysis is depicted in Table 11.

Table 11
QR code testing analysis on user usability

Application to open PDF Document	QR Code Scanner application	Advantages	Disadvantages
Browser	Smartphone Built-in scanner.	• Less percentage to zoom in on the document. • Readable QR code.	• The document file still needs to be zoom in.
Browser	Open source scanner.	• Readable QR code.	• More percentage to zoom in on the document.
Adobe PDF	Smartphone Built-in scanner.	• Readable QR code.	• More percentage to zoom in on the document.
Adobe PDF	Open source scanner.	• Readable QR code.	• More percentage to zoom in on the document.

According to Table 11, the properties of 'Application to open PDF Document' is the application used by the tester to open the digital document downloaded from the prototype system and the 'QR code Scanner application' is the application used by the tester to scan the QR code inside the digital document. From the analysis, it can be concluded that the QR code is readable but the distance to scan the QR code inside the document depends on the application to open the PDF document and the QR scanner application used. The analysis also indicates that the QR code inside the document is the best use with open the PDF document with a browser and scan the QR code using a smartphone built-in scanner as it resulted in less percentage to zoom in the PDF document. There are two parameters to consider when testing a steganography image: completeness of data and imperceptibility. In steganography image testing, the information obtained after scanning the QR code will be compared to the data from the system and database. Table 12 shows the results of the data comparison analysis.

Table 12
Steganography image testing analysis on data completeness

Properties	Data comparison	Advantages	Disadvantages
Accurate	✓	The data is accurate	The data provided is excessive.
Complete	✓	The data is complete	

Based on Table 12, it may be deduced that the information in the steganography image after the extraction process is complete and precise as the information in the QR code image but the drawback is the data provided might excessive which might reveal too much. The visibility testing on the steganography image is examined for the imperceptibility parameter, and the results of the analysis are shown in Table 13.

Table 13
Steganography image testing analysis on visibility

Property	Visibility	Advantages	Disadvantages
Visibility of the steganography image	The message cannot be seen but the carrier can be seen	• Contains ownership information. • Seen as a watermark image instead of a steganography image.	• May distract the content of the document. • Visible and can cause the viewer to focus on the image instead of the content.

Based on Table 13, the property is the steganography image visibility while the visibility is the visibility of the message from the steganography image. Generally, this evaluation can be concluded that the hidden message inside the steganography image is undetectable by the human eye but the carrier is still can be seen where it acts as a watermark and it may result in the steganography image can be removed.

Conclusions

It is essential to secure digital documents since it can assist in determining who distributed the document without the owner's consent. The QR code technology, as well as the steganography approach, are used in this work to provide copyright protection to the digital document downloaded from the prototype system. Multiple protection for the digital document in an E-Learning system is proposed. A QR code image, and a steganography image, which contains copyright information, along with the university's logo that acts as a watermark are embedded into the digital document. The proposed document copyright protection's effectiveness is demonstrated by experimental results; however, the digital document type used for the proposed idea is shown only on PDF type, author of the document shown only the name of the person who upload the document into the prototype, watermarking image and steganography image visibility shown visible and may distract content of the document, and the data hidden from the steganography image shown only can be identified on the researcher's side. As a result, ongoing research looks into ways to increase copyright protection by including other digital document types such as docx, pptx, and others, the best protection for the ownership of the document should be the author of the document, an invisible watermark image, and steganography image can be created, and a mechanism to obtain the steganography image from the document after the document is downloaded can be made to prove the ownership information on the user's side.

References

Rahman, M. A., Hassan, M., & Sabuddin, S. (2020). *COVID-19: Kecenderungan meneruskan penggunaan platform pembelajaran atas talian dalam kalangan guru pra perkhidmatan semasa Perintah Kawalan Pergerakan*. In Proceedings of the International Conference on Educational Research (InCER) (pp. 916–933).

Ahvanooey, M. T., Li, Q., Shim, H. J., & Huang, Y. (2018). A comparative analysis of information hiding techniques for copyright protection of text documents. *Security and Communication Networks, 2018*, 1–22.

Alajmi, M., Elashry, I., El-Sayed, H. S., & Faragallah, O. S. (2020). Steganography of encrypted messages inside valid QR codes. *IEEE Access, 8*, 27861–27873.

Amarendra, K., Mandhala, V. N., Gupta, B. C., Sudheshna, G. G., & Anusha, V. V. (2019). Image steganography using LSB. *International Journal of Scientific & Technology Research, 8*, 906–909.

Aru, O. E., & Ananaba, C. E. (2018). Detailed examination of information hiding techniques for copyright protection of text documents. *IOSR Journal of Applied Chemistry, 11*, 21–30.

Arkah, Z. M., Alzubaidi, L., Ali, A. A., & Abdulameer, A. T. (2020). Digital color documents authentication using QR code based on digital watermarking. In *Advances in Intelligent Systems and Computing* (Vol. 940, pp. 1094–1100).

Ashwini, C. M., Dipshikha, M. N., Vinay, V. K., & Kajal, S. P. (2021). A survey on novel approach for data hiding under QR code using visual secret sharing. *International Journal of Advance Scientific Research and Engineering Trends, 6*, 31–34.

Bao, W. (2020). COVID-19 and online teaching in higher education: A case study of Peking University. *Human Behavior and Emerging Technologies, 2*(2), 113–115. https://doi.org/10.1002/hbe2.191

Dang, Q. B., Louisa, K., Coustaty, M., Luqman, M. M., & Oqier, J. (2019). A blind document image watermarking approach based on discrete wavelet transform and QR code embedding. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)* (Vol. 8, pp. 1–6). IEEE.

Denso Wave Incorporated. (2021). *Error correction*. https://www.qrcode.com/en/about/error_correction.html

Dhawan, S. (2020). Online learning: A panacea in the time of COVID-19 crisis. *Journal of Educational Technology Systems, 49*(1), 5–22. https://doi.org/10.1177/00472395209340188

Hassan, A., & Hussein, A. (2020). Documents authentication and verification. In *IOP Conference Series: Materials Science and Engineering* (Vol. 765, pp. 1–10). IOP Publishing.

Hodges, C., Moore, S., Lockee, B., Trust, T., & Bond, A. (2020). The difference between emergency remote teaching and online learning. *Educause Review*. https://er.educause.edu/articles/2020/3/the-difference-between-emergency-remote-teaching-and-online-learning

Huang, H. C., Chen, Y. H., Chang, F. C., & Tseng, C. T. (2020). Multi-purpose watermarking with QR code applications. In *2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech)* (pp. 42–45). IEEE.

Kadhim, J. J., Premaratne, P., Vial, P. J., & Halloran, B. (2018). Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research. *Neurocomputing, 335*, 299–326.

Kumar, K. S., Kumar, C. M., Kumar, B. S., & Cristin, R. (2021). Highly imperceptible data hiding technique using MSB in the grayscale image. *Materials Today: Proceedings*.

Li, D., Gao, X., Sun, Y., & Cui, L. (2017). Research on anti-counterfeiting technology based on QR code image watermarking algorithm. *Journal of Information Hiding and Multimedia Signal Processing, 12*, 57–66.

Manimekalai, M., & Bakkiyalakshmi, R. (2017). Hide and seek: A new way to hide encrypted data in QR code using the concepts steganography and cryptography. *International Journal of Advanced Research in Computer and Communication Engineering, 6*, 538–540.

Subramanian, N., Elharrouss, O., Al-Maadeed, S., & Bouridane, A. (2021). Image steganography: A review of the recent advances. *IEEE Access, 9*, 23409–23423.

Thampi, S. M. (2014). Information hiding techniques: A tutorial review. *Cryptography and Security (cs.CR)*.

Thonky.com. (2021). *QR code tutorial*. https://www.thonky.com/qr-code-tutorial

UNESCO. (2021). *Education: From disruption to recovery*. https://en.unesco.org/covid19/educationresponse

Vaishnavi, D., & Subashini, T. S. (2015). Robust and invisible image watermarking in RGB color space using SVD. *Procedia Computer Science, 46*, 1770–1777.