Journal of Applied Engineering and Technological Science
Vol 7(2) 2026: 927-942

SEQURE: AN INTEGRATED DUAL DIGITAL QR AND INVISIBLE WATERMARK

P. Assiroj[1*], B. Hartati[2], Sohirin[3], B. Mulyawan[4], R. K. Astuti[5], I.A Prabadhi[6], C. Trinata[7], G.B Hertantyo[8], C. Susaningsih[9], M.F. Romdendine[10], O.P Martadireja[11]
Immigration Technology Management, Imipas Polytechnics Indonesia, Ministry of Immigration and Correctional Affairs, Republic of Indonesia, Indonesia[1,2,6,7,8,9,10,11]
Immigration Law, Imipas Polytechnics Indonesia, Ministry of Immigration and Correctional Affairs, Republic of Indonesia, Indonesia[3]
Immigration Administration, Imipas Polytechnics Indonesia, Ministry of Immigration and Correctional Affairs, Republic of Indonesia, Indonesia[4,5]
priati.assiroj@poltekim.ac.id*

Received: 11 November 2025, Revised: 30 December 2025, Accepted: 10 January 2026
**Corresponding Author*

ABSTRACT
This study introduces seQuRe, a novel approach that employs a dual-layer QR code system combined with invisible watermarking to enhance security measures in data transmission. The outer layer of the QR code facilitates general data accessibility while the inner layer, encrypted and embedded within the outer, secures sensitive information accessible only through specialized scanning. Utilizing advanced encryption standards (AES), this system ensures data integrity and confidentiality. The invisible watermarking further augments security by embedding additional data that verifies authenticity. Through systematic experimentation using Python and various libraries on Google Colaboratory, this experiment demonstrates the efficacy of seQuRe in resisting common cyber-attacks while maintaining data fidelity. We measured the peak-signal-to noise-ratio (PSNR) and the normalized cross-correlation (NCC) values of the QR images into which we had embedded watermarks, obtaining a PSNR value of 57.53 and an NCC value of 0.999. Subsequently, we also conducted simulation of attacks on the watermarked QR code with salt and pepper noise, speckle noise, and Gaussian noise attacks. From these attacks, we obtained PSNR values of 54.24 and NCC values of 0.6699 for the salt and pepper noise attack, 50.837 and 0.7319 for the speckle noise attack, and 33.17 and 0.0941 for the gaussian noise attack. The result underscores its potential application across industries requiring secure data handling and transmission. The implementation of such technology promises significant improvements in digital security, aiming to keep pace with the evolving landscape of cyber threats.
Keywords: *QR Code, Dual QR, seQuRe, PSNR, NCC*

1. Introduction

The implementation of QR codes has demonstrated significant benefits across various sectors, including healthcare, finance, and retail. Several studies highlight the success of utilizing QR codes in enhancing operations, such as inventory management in cafes (Fauziah et al., 2023), and improving payment systems in banking (Farrell et al., 2022), as well as facilitating secure data transmission in healthcare settings (Abdul-Jabbar & Farhan, 2023). Additionally, the use of QR codes has been expanded to include digital forensic applications for document verification and fraud detection (M. J. Tsai et al., 2023). Research on user acceptance in Sri Lanka has emphasized the importance of factors such as ease of use, perceived utility, and social influence in promoting QR code-based payment systems (Hewawasam et al., 2023). Overall, a comprehensive review of QR code implementation demonstrates its versatility and effectiveness in streamlining processes, enhancing security, and driving digital innovation across various industries, for labeling, information storage, and implementing interactive marketing strategies (Yao et al., 2022).

The Quick Response Code (QR Code) was introduced by Denso Wave in 1994 (Wave, 2024). This code can store various types of information such as text, URLs, contact details, and many more (Purdadi et al., 2023). These codes have been widely used in commercial tracking systems, marketing campaigns, and even contact tracing during the past COVID-19 pandemic (Wahsheh & S., 2022). QR codes are renowned for their high readability, large storage capacity,

and ease of scanning using smartphones with built-in scanning applications (Wahsheh & S., 2022), (Shokeen et al., 2022). Generally, the design of the QR code introduced by Denso is depicted in Figure 1 below.

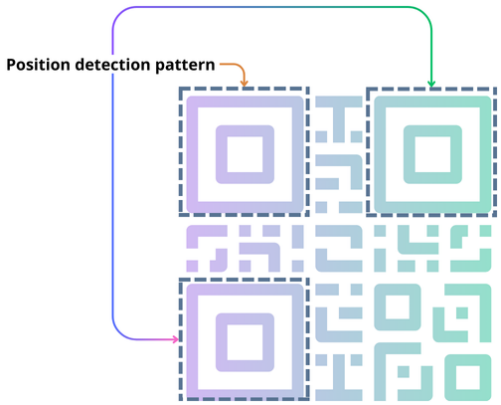

Fig. 1. Figure 1. QR code (Wave, 2024)

QR codes are characterized by three detection patterns located at three corners. These patterns are used by scanning devices to determine the orientation and position of the QR code. The detection patterns, which are large squares at the corners of the QR code, consist of a large outer square followed by a white square inside, and a smaller square at the center. These three squares help the scanning device to identify and adjust the position of the QR code. Details of the detection pattern sizes can be seen in Figure 2 below.

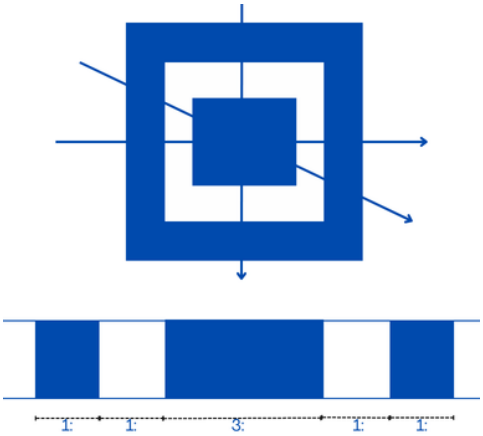

Fig. 2. Detection pattern on the QR code (Wave, 2024)

Horizontally, the detection pattern is created with an arrangement of blocks in a size ratio of 1:1:3:1:1. This sequence is used by scanners to recognize the pattern and confirm that it is part of QR code, regardless of the QR code's orientation position (Gu et al., 2011). Deviations from the standard size ratio of the detection pattern can lead to issues such as difficulty in recognizing the QR code, reading errors, decreased accuracy, inability to scan from various angles, and compatibility issues, given that QR codes are an international standard. Creating QR codes involves various rules and considerations. QR codes are two-dimensional matrix images used to store and quickly decrypt information, particularly with handheld devices such as smartphones (Shokeen et al., 2022). These codes can be encrypted with user data, as seen in the application of QR code on Trusmi batik patterns for online trading, requiring legal protection through policy regulations (Priowirjanto et al., 2022). QR codes also feature an error correction mechanism that allows for data recovery among damaged modules. There are four error correction levels in QR codes, L for low, M for medium, Q for quartile, and H for high, with error tolerances of 7%, 15%, 25%, and 30% respectively (Chow et al., 2020).

2. Literature Review

Various techniques have been proposed to enhance the identification rate of QR codes by addressing disturbances such as uneven backgrounds, image distortion, and noise (Latha & Rao, 2023), (Dr Girisha H et al., 2022). Additionally, innovative approaches such as integrating duplicate detection patterns into QR codes have been developed to address the growing issues of counterfeiting in product security (Picard et al., 2021), and algorithms like DWT-QR, which utilize wavelet transformations, have proven to enhance the visual quality and resilience of QR codes against attacks (M.-J. Tsai & Hsieh, 2019). Currently, QR codes are primarily used as tools for conveying messages or information, incorporating identifiers as demonstrated in the following example figures.

Fig. 1. QR code 1 sample

Fig. 2. QR code 2 sample

Fig. 3. QR code 3 sample

Fig. 4. QR code 4 sample

The four example images mentioned above can be freely accessed on the internet. Figure 3 is an example of a QR code usage where the owner's signature is embedded in the center. Figure 4 illustrates a QR code used to store the address of a website, while Figure 4 is an example of a QR code storing general information from the owner in the center. Figure 6 depicts a QR code as a verification tool, displaying verifier information in the middle of the QR code. The examples in Figures 3, 4, and 5 represent common uses of QR codes, while the example in Figure 6 illustrates a specialized use of QR codes. This specialization involves embedding critical information that can only be scanned by specific scanners. However, all this critical information is encapsulated within an outer QR code that, in practice, should only be used to insert public information.

Despite this, concerns about QR code security are increasingly prominent, leading to the exploration of various encryption methods, such as cryptography, to enhance their protection (Wahsheh & S., 2022). An innovative scheme has been proposed, consisting of a three-tier QR code system based on super-pixel segmentation, to enhance security and resistance to disturbance (Wu et al., 2022). QR codes have diverse applications across multiple industries. This two-dimensional code is widely used in commercial activities, high technology, storage, transportation, wholesale, and retail industries for cost-effective and rapid labeling of goods (Yao

et al., 2022). In the audiovisual sectors, QR codes can be integrated with television programs to enhance interactivity, engagement, and viewer participation, serving as a bridge between traditional television audiences and digital media platforms (Gallardo-Camacho & Melendo-Rodríguez-Carmona, 2023). QR codes are also used for data protection and concealing sensitive information, with proposed encryption systems aimed at enhancing security and privacy in information transmission (Al Dallal & Al Mukhtar, 2023).

Implementing two-factor authentication algorithms and encryption techniques can enhance data security and protect against the theft of important information. In wireless communication, QR codes are also used for efficient data transfer devices, accompanied by robust security measures (Basherlou et al., 2023). To address security challenges posed by quantum technology, Quantum safe QR code with lattice-based cryptography have been proposed to ensure anti-counterfeiting and anti-tampering features, securing data integrity and authenticity in the post-quantum era (Lou et al., 2023). Therefore, the combination of secure application architecture, encryption methods, and post-quantum cryptography can significantly enhance QR code security and protect users from potential cyber threats. At the advance level, we have implemented this concept in seQuRe, which uses two layers of QR codes, namely outerQR code and innerQR code. The outerQR code is designed like a standard QR code and only stores public information accessible to anyone who scans it. Figure 7 below illustrates the outerQR that we have generated.

Fig. 7. Outer QR code

Figure 7 is the first layer or outer layer of the seQuRe QR code. This QR code contains general information that is accessible to everyone and features a white box in its center designed to incorporate a second, inner-layer QR code. This innerQR code is smaller in size and encrypted using an advanced encryption standard (AES) method to store critical data and information. The InnerQR code is depicted in Figure 8 below.

Fig. 8. Inner QR code

Figure 8 is the inner QR code containing encrypted information or data. This is a smaller QR than the outer QR code to be inserted into the center of the outer QR code as shown in Figure 9, and this is seQuRe looks like.

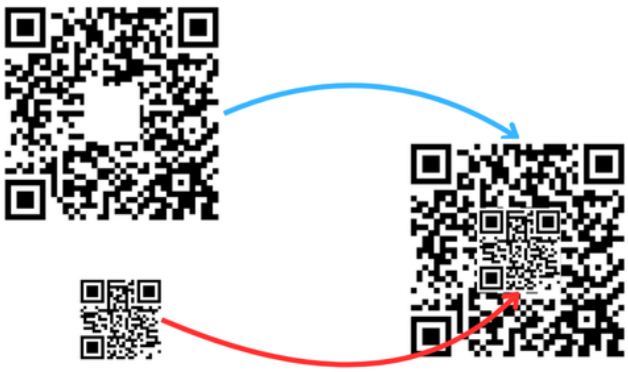

Fig. 9. seQuRe

A QR code image is inserted into a white box in the center of an outerQR code. With these two layers, seQuRe offers two levels of security. The first level is general information that can be accessed by anyone with QR scanner, and the second level consists of sensitive data or information protected by encryption and accessible only with a special scanner equipped with the appropriate decryption key. Encryption techniques play a crucial role in securing QR codes by ensuring the confidentiality, privacy, and integrity of the information being transmitted. For example, the use of visual cryptography schemes and the AES algorithm can be employed to encrypt QR codes (Goon et al., 2023), as well as the utilization of encryption methods such as symmetric one-time pad (OTP) (Malallah et al., 2023). These encryption techniques not only protect the content of the QR code but also preserve the physical appearance and heterogeneity of the code during the encryption and decryption process. This ensures the integrity and security of the data (Iliyasu & Iliyasu, 2022), (Zhang et al., 2022). Advanced Encryption Standard (AES) is widely used symmetric block cipher algorithm that ensures a high level of data security (Wade, 2023). AES operates by converting the input image into a binary format, generating a random 128-bit key, and creating subkeys for each encryption round. The binary images is then divided into 128-bit blocks, and the encryption algorithm is applied using the subkeys, ensuring secure encryption and protection against unauthorized access (S et al., 2023).

AES is a versatile encryption technique suitable for a wide range of applications, including online banking and internet of things (IoT) devices (Mrs. M. Saritha et al., 2023), (Gavaskar et al., 2022). The key-dependent and dynamic nature of the algorithm's step enhances security, ensuring that only authorized parties with the decryption key can access the encrypted data. With key processes such as SubByte, ShiftRows, MixColumns, and AddRoundKey, AES ensures a high level of security and effectiveness in both encryption and decryption procedures (Taufiqqurrachman & Elsandi, 2022). AES provides a dynamic and secure encryption process, making it difficult for hackers to decrypt data without the correct key. This ensures that important data and sensitive information are protected from unauthorized access (Shete & Kohle, 2022). By utilizing AES, data encryption and decryption during file transfer process are safeguarded, preventing hackers from intercepting and exploiting files during transmission (Rani et al., 2023). The Elliptic Curve Diffie-Hellman (ECDH) encryption algorithm is a secure method for key exchange in wireless sensor networks (Abood et al., 2022). This technique involves the use of elliptic curve cryptography (ECC) to ap message characters to points on the curve, enhancing efficiency by eliminating the complexity of encoding. The security strength of the algorithm is directly proportional to the key length, making brute-force attacks nearly impractical (Kodali & Sarma, 2014). Combining ECDH with rivest-shamir-adleman (RSA) enhance security, enabling secure and efficient data transmission (Abood et al., 2022). The use of ECC algorithms ensures the creation of strong secret keys that are resistant to man-in-the-middle attacks in cryptographic applications, such as smartcards and wireless networks (Banerjee & Patil, 2018). The ECDH protocol ensures secure key exchange in communications by leveraging the efficiency and security of ECC. ECDH is more efficient that traditional techniques such as RSA, offering smaller key sizes and simpler computational requirements (Abusukhon et al., 2019). Various developments have been implemented to address security challenges in communication, such as

the unbalanced ECDH (UECDH) scheme. This scheme is designed to prevent man-in-the-middle attacks and duplication, ECDH is also used in securing secret key exchanges on wireless sensor devices, proving to be different, optimal in resource usage, and capable of enhancing performance by reducing the complexity of scalar multiplication (Teguig et al., 2017).

Watermarking using the Least Significant Bit (LSB) technique is a method of hiding secret messages within digital data to protect information and ensure its authenticity (Widiyono et al., 2022), (Faheem et al., 2023). This technique involves replacing the least significant bits of pixel values in an image with secret data, allowing for invisible changes to the visual appearance of the image while enabling the extraction of the hidden message when needed. The LSB technique is widely used because it is simple, offers high payload capacity, and maintains the integrity of the original image. With LSB, we can embed secret information or data in images without altering the overall image quality. The LSB method has proven effective in various applications, such as securing audio files in digital images, and provides resistance to image processing and geometric attacks, making it crucial in digital data protection and authentication techniques (Sakshi et al., 2022). Another development in LSB techniques is quantum image watermarking (Khairunnisa et al., 2022). This method is used to embed watermark images for color images and is implemented to ensure copyright protection by generating an invisible watermark that is resistant to noise attack. The LSB technique allows for direct pixel manipulation and the embedding of hidden data without significantly altering the appearance if the main image (Miftahul Amri et al., 2023). Although effective at hiding watermarks, LSB is considered fragile against attacks compared to techniques like the Discrete Wavelet Transform (DWT) (Makhrib & Abdulamir, 2022). However, LSB is also known for its watermark being difficult to detect, and the embedded information not being easily extracted (Gupta et al., 2022). The advantage of the LSB technique lies in its high payload capacity and direct pixel manipulation capabilities. Additionally, LSB is suitable for hiding various types of data, such as text or grayscale images, in digital images and videos (Mouhsen & Hussain, 2022). LSB maintains the position of the watermark during geometric attacks by identifying embedding locations based on the image gradient (Faheem et al., 2023). LSB has proven to be an effective method for securely embedding watermarks in digital media while preserving the quality and integrity of the image.

3. Research Methods

At seQuRe we create two types of QR codes that we refer to as the outerQR code and the innerQR code. Both QR codes are designed to work together within a system that aims to segregate information into general and specific categories. The outerQR code is placed on the exterior with an empty box in the middle, which is filled with the innerQR code. This outerQR code contains general information that is accessible to everyone and can be scanned by various types of QR code scanners. Examples of general information that can be included in the outerQR code are not limited to product identity, contact information, or links to official websites. Its function is to provide quick and easy access for users to non-sensitive and general information. This allows users to easily verify the authenticity of a product or obtain basic information.

On the other hand, the innerQR is placed within the empty box at the center of the outerQR. This QR contains specific and sensitive information that can only be accessed by certain parties with specialized scanners. This arrangement enhances data security. The composite QR code, resulting from the combination of the outerQR and innerQR, which we call seQuRe, appears as a single unified code but actually consists of two distinct layers of information. To conceal data within the innerQR we combine AES encryption techniques and ECDH to enhance the security of the encrypted data. The use of AES, known for its speed and efficiency in symmetric encryption, provides a strong layer protection for the data. Meanwhile, ECDH offers a secure key exchange method through public key cryptography, allowing two parties to securely generate an encryption key that is accessible only to them. This combination ensures that the data is not only encrypted securely but also that the key exchange process is conducted with maximum security, reducing the risk of interception by unwanted third parties (Barker et al., 2007).

The implementation of AES and ECDH provides advantages in terms of scalability and flexibility of the security system. By leveraging the strengths of each algorithm, the system can be tailored to meet various security and performance needs. Recommendation for the key

exchange scheme should consider factors of security and efficiency, where the combination of ECDH and AES offers an optimal balance between these factors (Barker et al., 2013). The seQuRe application enables differentiation between publicly accessible information and information that can only be accessed by specific parties, thereby enhancing overall security. It also provides flexibility in the way information is presented and accessed, allowing various organizations to convey information efficiently without compromising security. The practical applications of this method are extensive. For instance, in the pharmaceutical industry, the outerQR can be used to provide basic information about drug, such as its composition and usage instruction, while the innerQR can store confidential data like the production batch number and manufacturing details, accessible only to authorized personnel. In the logistic sector, the outerQR can display general tracking information accessible to customers, while the innerQR holds detailed shipping details and specific instructions relevant to logistic staff. This innovative approach enhances both security and efficiency of information verification.

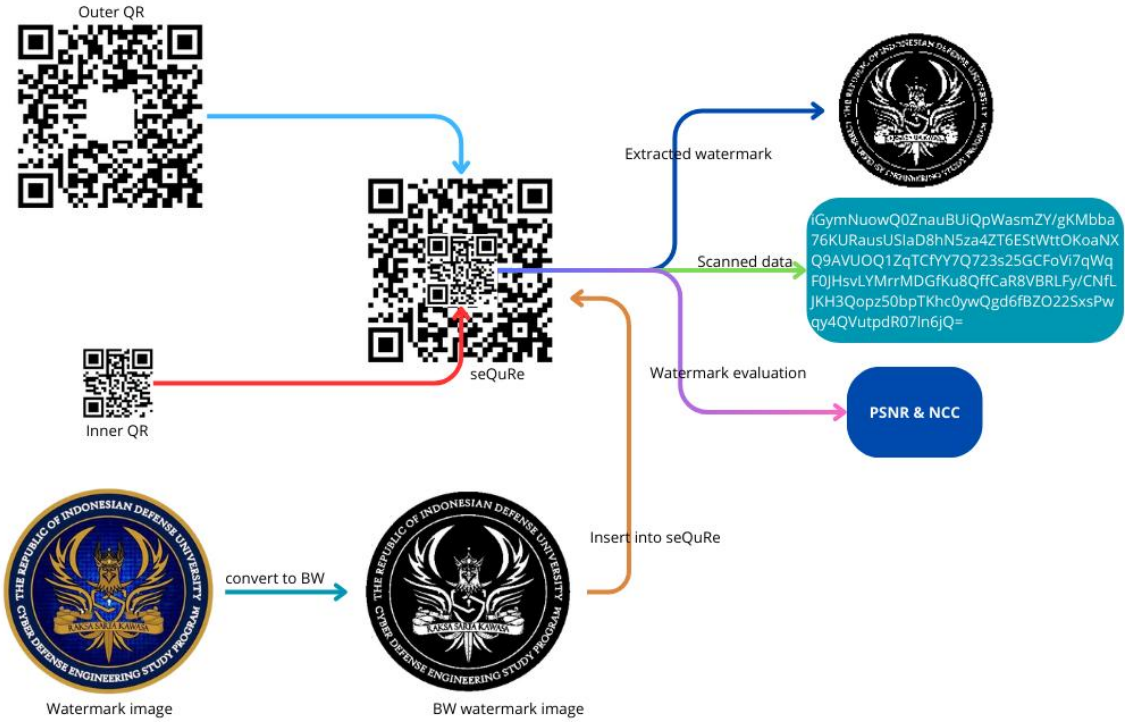

Fig. 10. seQuRe and watermark embedding

Figure 10 above outlines the systematic steps in generating seQuRe, a dual digital QR code with encryption and the insertion of an invisible watermark. To maintain the authenticity of the QR code, we embed an invisible watermark using LSB technique. We first convert the watermark into black and white to facilitate the integration process. This conversion not only makes it compatible with the seQuRe format, which generally operates in binary, but also optimizes the contrast and clarity of the data when scanned. After the watermark image is converted to black and white, we insert it into innerQR. This insertion involves encoding the image data into the patterns of seQuRe so as not to disrupt the visual and data within seQuRe, and it can be extracted again.

Once the watermark is successfully embedded, seQuRe is ready to be scanned. The extraction of the watermark provides additional proof of the authenticity of integrity of the encoded data or information, as the watermark must match specific criteria agreed upon by the sender and receiver of the information. The encryption performed provides an additional layer of security, ensuring that the data is not easily read or altered without proper authorization. This technique is extremely useful in securing documents or digital data across various applications. Hsu & Wu (1999) proposes a method to embed digital watermarks in images, effectively

preventing unauthorized copying and attesting the origin of the images, even after processing, cropping, and JPEG lossy compression. Pan et.al (Pan et al., 2021) employ improved DWT-SVD and SMS algorithms to improve digital watermarking on QR codes, resulting in higher peak signal-to-noise ratio (PSNR) and practicable protection of information. A survey on various aspects of watermarking has been conducted by (Kumar et al., 2020). This survey highlights its importance for copyright protection, content authentication, and identity theft prevention in various multimedia and database applications.

4. Results and Discussions

All experiments were conducted on the Google Colaboratory platform using the Python programming language and a standard runtime. We utilized several libraries related to encryption, QR code generation, and image processing. These libraries are json, pycryptodome, qrcode, pyqrcode, PIL, base64, and numpy.

We use the 'json' library to generate data in JavaScript Object Notation (JSON) format. The 'json' module supports parsing from strings or files and converting them into Python data structures like dictionaries, and vice versa (*Json — JSON Encoder and Decoder — Python 3.12.4 Documentation*, 2024). The 'Crypto.PublicKey.ECC' library, part of PyCryptodome package, provides tools for encryption, decryption, and key management using ECC. ECC is used as a secure and efficient encryption method, especially in applications with limited resources. Similarly, 'Crypto.Cipher.AES' is also part of PyCryptodome and is used for encryption and decryption operations using AES. The 'Crypto.Hash.SHA256' library, another component of PyCryptodome, is used to generate hash values using the SHA-256 algorithm, which is commonly used to verify data integrity. The function 'get_random_bytes' from PyCryptodome is used to generate cryptographically secure random bytes and is utilized for generating secure keys (*Pycryptodome · PyPI*, 2024). The base64 module provides functions for encoding and decoding data in Base64 format. This method is commonly used to encode binary data for transfer or storage in plain text (*Base64 — Base16, Base32, Base64, Base85 Data Encodings — Python 3.12.4 Documentation*, 2024). The 'qrcode' library is employed to create QR codes that can store data or information in a format easily scannable (*Qrcode · PyPI*, 2023). A fork of the Python image library (PIL) is used to open, manipulate, and save various image formats (*Pillow (PIL Fork) 10.4.0 Documentation*, 2023), while 'numpy' is utilized for its support of large-sized arrays (*Numpy · PyPI*, 2023) (Harris et al., 2020). The 'google.colab.files' library serves as a means to upload files during the experimental process (Google.com, 2022). The file we upload are watermark images, examples of which can be seen in Figure 11.

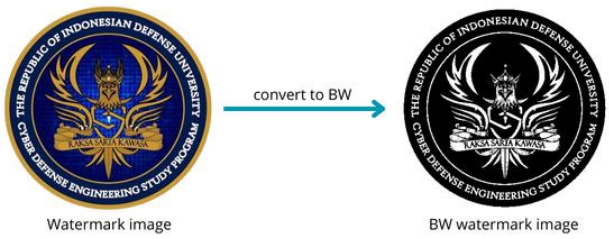

Fig. 11. Watermark image

The successfully uploaded watermark image then converted into black and white. We developed a function that can transform color images into black-and-white images, but first, we convert them to grayscale. Once the images are in grayscale, we convert them into black and white. Subsequently, we save these as new files that we embed as watermarks. The outer QR code is created using the 'qrcode' library with slight modifications to generate the appropriate QR code. We employ a high correction level to prevent damage to the QR code that is successfully generated. At this step, we also encode data to be included in the QR code. The data we use as a sample is formatted in json, and can be seen in Figure 12 below.

```
data = {
    "ORIGIN": "REPUBLIC OF INDONESIA",
    "NAME": 'THE REPUBLIC OF INDONESIA DEFENSE UNIVERSITY',
    "FACULTY": "FACULTY OF DEFENSE SCIENCE AND ENGINEERING",
    "PROGRAM": "CYBER DEFENSE ENGINEERING",
    "WEBSITE": "https://fstp.idu.ac.id/rps",
    "CONTACT": "rps@idu.ac.id"
}
```

Fig. 12. Public data sample

The sample information as shown above, is general, accessible to everyone, and includes details such as origin, name of the institution, faculty, program, website, and contact information. This information is typically intended for public dissemination and does not contain elements that could directly compromise the privacy or security of the associated institution or individuals. Once the data is successfully encoded into a QR code, we save this QR code as outer QR code. Figure 13 below shows the successfully generated outer QR code.

Fig. 13. Generated outer QR

In the subsequent process, we utilize ECC and AES to generate key pairs and to encrypt and decrypt data. We use ECC to generate the keys, specifically employing the ‘P-256’ curve, which is one of the various curves used in ECC. From this private key, a public key is generated, which can then be used for cryptographic purposes such as encryption. AES is a symmetric encryption standard, meaning the same key is used for both encrypting and decrypting data. Data is encrypted using the EAX mode of AES, an operational mode that also ensures data integrity. The encryption result is then combined with a ‘nonce’, a value used only once to ensure high security. Finally, all data is encoded into the base64 format. The functions used can be seen in the Figure 14.

```
# Generate ECC key pair
private_key = ECC.generate(curve='P-256')
public_key = private_key.public_key()

# Function to encrypt data with AES using a symmetric key
def encrypt_data(data, key):
    cipher = AES.new(key, AES.MODE_EAX)
    nonce = cipher.nonce
    ciphertext, tag = cipher.encrypt_and_digest(data.encode('utf-8'))
    return base64.b64encode(nonce + ciphertext).decode('utf-8')

# Function to decrypt data with AES using a symmetric key
def decrypt_data(encrypted_data, key):
    encrypted_data = base64.b64decode(encrypted_data.encode('utf-8'))
    nonce = encrypted_data[:16]
    ciphertext = encrypted_data[16:]
    cipher = AES.new(key, AES.MODE_EAX, nonce=nonce)
    plaintext = cipher.decrypt(ciphertext)
    return plaintext.decode('utf-8')

# Generate a random symmetric key for AES
symmetric_key = get_random_bytes(16)
```

Fig. 14. Encryption and decryption

The decrypt function shown in Figure 15 performs the reverse process. Encrypted data received in base64 format is decoded. The nonce used during the encryption process is separated from the encrypted text, and both are utilized to decrypt the text to restore it to its original format. The random symmetric key generated by AES in a 16-byte binary form used in both the encryption and decryption functions must be securely stored and shared only among authorized parties who have access to the encrypted data. The data we use as an example is student data from the Cyber Defense Engineering program. This data can be modified by using other more confidential or sensitive data. After successfully encrypting the data, we create a smaller QR code to accommodate the encrypted data or information. We also use a high level of error correction to prevent damage to the successfully generated smaller QR code. We refer to this smaller QR code as innerQR code, and its visual is shown in Figure 15.

Fig. 15. Inner QR code

Up to this point, we have created two QR codes, outerQR and innerQR code, each containing data or information. The watermark image embedded is a black and white type. This choice facilitates the embedding process, as only black and white values need to be processed in the watermark. For the embedding process, the size of the watermark image is adjusted to match the size of the innerQR. Both the innerQR image and the appropriately sized watermark are then converted into arrays using the ‘numpy’ library. This conversion is conducted to facilitate pixel manipulation via direct array operations. Subsequently, embedding using the LSB technique is performed on the innerQR code to include an invisible watermark without disturbing the visual appearance of the QR code. The result is a QR code that looks exactly like the innerQR shown in Figure 16. Both QR codes are then combined to create ‘seQuRe’, a composite QR code from the outerQR and innerQR with an invisible watermark, which can be used externally to ensure the

authenticity of a product. The presence of the invisible watermark serves as an anti-counterfeiting feature. Figure 16 illustrates the visual form of seQuRe.

Fig. 16. Generated seQuRe

We evaluate the watermark using the peak-signal-to-noise ratio (PSNR) to measure the quality of the images. A high PSNR value suggests good image quality with minimal noise (Kumar et al., 2020). We establish that the maximum pixel value for each image is 255. NCC measures the similarity between a template and a portion of an image by comparing their pixel values. NCC values are crucial for finding image correspondence accuracy (Xuan et al., 2022). Recent advancements have enhanced the robustness of NCC through processing images with 'siamese' convolutional networks, which optimize the contrast between NCC values of true and false matches, leading to a significant reduction in false matches (Rajeswari et al., 2013). This improvement is particularly valuable for applications like connectomics, where billions of template matches may be needed to assemble 2D images into a 3D image stack. The enhanced accuracy achieved by leveraging NCC values is beneficial not only for connectomics but also for various other computer vision applications relying on template matching. A high NCC value approaching 1 indicates a high similarity between the original watermark and the extracted one. This implies that the watermark can withstand certain types of distortions. This work obtains the PSNR and NCC as depicted in Figure 17 below.

```
print(f"PSNR: {psnr_value}")
print(f"NCC: {ncc_value}")


PSNR: 57.53684854040909
NCC: 0.9999999999999999
```

Fig. 17. PSNR and NCC value of invisible watermark

The obtained PSNR values indicate a relatively high level, meaning that the difference between the original image and the watermarked image is very small. A higher PSNR signifies that the watermarked image closely resembles the original in terms of visual quality. PSNR values above 40 are considered excellent in the context of watermarking because the changes included by the embedding of the watermark are nearly imperceptible to the human eye. Meanwhile, the NCC value, which measures the similarity between two images or signals, ranges from -1 to 1, where a value of 1 indicates perfect similarity, 0 indicates no correlation, and -1 indicates a perfect negative correlation. An NCC value of 0.99, being very close to 1, suggests that the original and watermarked images are nearly identical in content and structure. This indicates that the watermark embedding process was successfully conducted without disrupting the original image. Our evaluation involves subjecting images embedded with watermarks to attacks such as salt and pepper noise, speckle noise, and Gaussian noise attack. These attacks simulate real-world conditions to evaluate the watermark's robustness and resilience. All these attacks are measured

using the normalized cross-correlation (NCC) value (Abdulwahed & Ahmed, 2020). We effectively measure the similarity between two images by quantifying the correlation between their pixel intensities while taking into account their mean and standard deviation. Salt and pepper noise refers to a type of noise that is added to an image during the watermark process. This noise is characterized by randomly occurring white and black pixels, resembling grains of salt and pepper scattered throughout the image. The PSNR and NCC values for each attack can be seen in Table 1.

Table 1 – PSNR and NCC values for each attack

Attacks	PSNR value	NCC value
Salt and Pepper Noise	54.24	0.6699
Speckle Noise	50.837	0.7319
Gaussian Noise	33.17	0.0941

Salt and pepper noise attack can be intentionally introduced to the image in an attempt to disrupt or obscure the embedded watermark. The obtained PSNR and NCC values are relatively high, indicating that the image remains resilient against these attacks. However, the NCC values show a significant reduction in similarity from the original image, suggesting that with these attacks, the watermark becomes difficult to recognize. Then the next attack we have conducted is adding speckle noise attack. This is a type of noise characterized by random variations in intensity or brightness within an image. Speckle noise can be intentionally added to the image to disrupt the embedded watermark and make it more difficult to extract or detect. This type of noise can affect the integrity and visibility of the watermark. Based on the PSNR and NCC values, it can be determined that this attack has a relatively milder impact on the image.

The last attack we have conducted is gaussian noise attack. This attack introduces a normal probability distribution to each pixel, which is randomly distributed yet follows a normal distribution pattern (Abdulwahed. & Ahmed, 2020), (Khan & Goyal, 2016). The gaussian noise attack causes changes in pixel intensities that are randomly distributed, resulting in a fine grainy effect evenly spread across the entire image. The obtained PSNR and NCC values are lower than those of the original watermark, indicating that the gaussian noise attack cause significant degradation in image quality. The low NCC values also suggest that with the gaussian noise attack, the watermark becomes difficult to extract. We concluded the experiment by decrypting the data contained within the seQuRe and extracting watermark. This was done to ensure that all experiments were conducted in alignment with the intended objectives. The results of the data decryption are displayed in Figure 18, and the watermark extraction results are shown in Figure 19.

```
Common Information:  {'ORIGIN': 'REPUBLIC OF INDONESIA', 'NAME': 'THE REPUBLIC OF INDONESIA DEFENSE
Confidential:  {'STUDENTID': '120230405011', 'ID NUMBER': '32156656518278xxxx', 'DOB': '01/01/2003'
```

Fig. 18. Decryption results

Fig. 19. Extracted watermark

The decryption results displayed in Figure 18 demonstrate that the encryption and decryption of data encoded in the seQuRe were successfully performed. Meanwhile, the watermark extraction results shown in Figure 19 indicate a similarity with the black and white watermark image embedded in the innerQR code. This can serve as a tool to verify the authenticity of the seQuRe or as an anti-copying feature.

5. Conclusion

The exploration of seQuRe underscore its pivotal role in enhancing security in digital communication through the integration of dual-layer QR codes and invisible watermarking. This technology not only facilitates robust encryption mechanism but also supports authentication and integrity verification across diverse applications. Its application spans various domains from healthcare to retail, proving its versatility and effectiveness. The incorporation of advanced encryption standards like AES, along with elliptic curve cryptography, provides a strong defense agains potential cyber threats, making seQuRe as a promising solution for securing digital transactions and data exchanges. Future work could further refine these techniques, expand their applicability, and explore new cryptographic challenges in an increasingly digital world.

References

Abdul-Jabbar, S. S., & Farhan, A. K. (2023). Secure QR-Code Generation in Healthcare. *Karbala International Journal of Modern Science*, *9*(2), 307–315. https://doi.org/10.33640/2405-609X.3294

Abdulwahed, M. N., & Ahmed, A. Kamil. (2020). Improved anti-noise attack ability of image encryption algorithm using de-noising technique. *Telkomnika (Telecommunication Computing Electronics and Control)*, *18*(6), 3080–3087. https://doi.org/10.12928/TELKOMNIKA.v18i6.16384

Abood, B., Faisal, A. N., & Hamed, Q. A. (2022). Data transmitted encryption for clustering protocol in heterogeneous wireless sensor networks. *Indonesian Journal of Electrical Engineering and Computer Science*, *25*(1), 347–357. https://doi.org/10.11591/ijeecs.v25.i1.pp347-357

Abusukhon, A., Mohammad, Z., & Al-Thaher, A. (2019). Efficient and Secure Key Exchange Protocol Based on Elliptic Curve and Security Models. *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*. https://doi.org/https://doi.org/10.1109/JEEIT.2019.8717496

Al Dallal, H. R. H., & Al Mukhtar, W. N. M. (2023). A QR Code Used for Personal Information Based on Multi-Layer Encryption System. *International Journal of Interactive Mobile Technologies*, *17*(9), 44–56. https://doi.org/10.3991/ijim.v17i09.38777

Banerjee, S., & Patil, A. (2018). ECC Based Encryption Algorithm for Lightweight Cryptography. In A. Abraham, A. K. Cherukuri, P. Melin, & N. Gandhi (Eds.), *International Conference on Intelligent Systems Design and Applications*. Springer, Cham. https://doi.org/https://doi.org/10.1007/978-3-030-16657-1_56

Barker, E., Chen, L., Roginsky, A., & Smid, M. (2013). *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)*. *NIST Special Publication 800-56A*, 1–114.

Barker, E., Johnson, D., & Smid, M. (2007). *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)*. *NIST Special Publication 800-56A*, 1–114.

base64 — Base16, Base32, Base64, Base85 Data Encodings — Python 3.12.4 documentation. (2024). https://docs.python.org/3/library/base64.html

Basherlou, H. J., Ullah, A., Parchin, N. O., See, C. H., & Abd-Alhameed, R. A. (2023). QR-Code Pixelated Antenna with Multi-Factor Authentication for Wireless and Security Applications. *2023 First International Conference on Microwave, Antenna and Communication (MAC)*. https://doi.org/10.1109/MAC58191.2023.10177088

Chow, Y. W., Susilo, W., Baek, J., & Kim, J. (2020). QR Code Watermarking for Digital Images. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial*

Intelligence and Lecture Notes in Bioinformatics), *11897 LNCS*(January), 25–37. https://doi.org/10.1007/978-3-030-39303-8_3

Dr Girisha H, A Dheerendra Kumar, Atithi Singh, Bharath K P, & Deepak. (2022). QR Code Detection. *International Journal of Advanced Research in Science, Communication and Technology*, *2*(9), 357–360. https://doi.org/10.48175/ijarsct-5353

Faheem, Z. Bin, Ishaq, A., Rustam, F., De, I., Díez, T., Gavilanes, D., Vergara, M. M., & Ashraf, I. (2023). Image Watermarking Using Least Significant Bit and Canny Edge Detection. *Sensors (Switzerland)*, 1–15. https://www.mdpi.com/1424-8220/23/3/1210

Farrell, M., Ikhwan, A. D., Kartika, W., Rahadi, R. A., Azkaenza, M., & Haq, M. A. (2022). Implementation Study of Quick Response Code Indonesia Standard (QRIS) in Papua Province. *Jurnal Manajemen Indonesia*, *22*(3), 289. https://doi.org/10.25124/jmi.v22i3.4025

Fauziah, A. N., Rasyid, A., & Wijayanti, R. A. (2023). Implementation of Data Collection and Payment Control Systems Automatically Using Android -Based QR Code. *Jartel*, *13*(1), 95–102. https://doi.org/10.33795/jartel.v13i1.521

Gallardo-Camacho, J., & Melendo-Rodríguez-Carmona, L. (2023). The use of QR codes to fuel transmedia strategy in the ecosystem of audiovisual media groups. *Profesional de La Informacion*, *32*(2), 1–9. https://doi.org/10.3145/epi.2023.mar.16

Gavaskar, K., Ragupathy, U. S., Ravivarma, G., & Priyadharshan, P. S. (2022). AES Algorithm using Dynamic Shift Rows, Sub Bytes and Mix Column Operations for Systems Security wih Optimal Delay. *Wireless Personal Communications*, 1–14. https://doi.org/10.21203/rs.3.rs-1973978/v1

Google.com. (2022). *Google Colaboratory*. https://colab.research.google.com/

Goon, S., Pal, D., Dihidar, S., & Roy, S. (2023). *QR Code-Based Digital Payment System Using Visual Cryptography*. 145–158. https://doi.org/10.1007/978-981-99-0550-8_11

Gu, Z., Scott, M. R., Chen, G., & Tien, J. Y. (2011). *QR CODE DETECTION* (US20110290882A1). US Patent Application Publication.

Gupta, M. K., Dadheech, P., Kumar, A., Dogiwal, S. R., Poonia, R. C., Raja, L., & Bhatt, D. P. (2022). Detection and localization for watermarking technique using LSB encryption for DICOM Image. *Journal of Discrete Mathematical Sciences and Cryptography*, *25*(1), 193–204. https://doi.org/10.1080/09720529.2021.2009193

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/S41586-020-2649-2

Hewawasam, P. C., Jaharadak, A. A. Bin, Khatibi, A., & Azam, S. M. F. (2023). QR Code Enabled Payment Solutions in Creating a Cashless Society among Sri Lankan Consumers—A Literature Review. *Journal of Service Science and Management*, *16*(02), 110–132. https://doi.org/10.4236/jssm.2023.162008

Hsu, C.-T., & Wu, J.-L. (1999). Hidden Digital watermarks in Images. *Proceedings of the International Conference on Wavelet Analysis and Its Applications (WAA)*, *8*(1), 58–68. https://doi.org/10.1142/9789812796769_0155

Iliyasu, A. M., & Iliyasu, A. M. (2022). Securing QR Codes Using a Hybrid Pseudo Baker's Mapped Cellular Automation. *2022 IEEE 8th International Conference on Cloud Computing and Intelligent Systems (CCIS)*. https://doi.org/10.1109/CCIS57298.2022.10016396

json — JSON encoder and decoder — Python 3.12.4 documentation. (2024). https://docs.python.org/3/library/json.html#

Khairunnisa, M., Budiman, G., & Novamizanti, L. (2022). Quantum Image Watermarking Based on Least Significant Bit for Color Images. *2022 IEEE Symposium on Future Telecommunication Technologies (SOFTT)*. https://doi.org/10.1109/SOFTT56880.2022.10010071

Khan, M. R., & Goyal, A. (2016). Gaussian Noise attack A nalysis of Non Blind Multiplicative Watermarking using 2D-DWT. *International Journal of Computer Science and*

Information Technologies, *7*(6), 2481–2486.

Kodali, R. K., & Sarma, N. V. S. N. (2014). Energy Efficient ECC Encryption Using ECDH. *Emerging Research in Electronics, Computer Science and Technology*. https://doi.org/10.1007/978-81-322-1157-0_48

Kumar, S., Singh, B. K., & Yadav, M. (2020). A Recent Survey on Multimedia and Database Watermarking. *Multimedia Tools and Applications*, *79*(27–28), 20149–20197. https://doi.org/10.1007/s11042-020-08881-y

Latha, Y. M., & Rao, B. S. (2023). Advanced Denoising Model for QR Code Images Using Hough Transformation and Convolutional Neural Networks. *Traitement Du Signal*, *40*(3), 1243–1249. https://doi.org/10.18280/ts.400342

Lou, S., Shen, W., Shen, G., Cui, L., Yang, F., Deng, J., & Lyu, S. (2023). QR code anti-counterfeiting technique with lattice-based cryptography. *International Conference on Cyber Security, Artificial Intelligence, and Digital Economy (CSAIDE 2023)*. https://doi.org/10.1117/12.2681559

Makhrib, Z. F., & Abdulamir, A. (2022). A Hybrid Digital Image Watermarking By Using DWT and LSB Method. *Iraqi Journal of Computer, Communication, Control and System Engineering*, *22*(4), 115–126. https://doi.org/10.33103/uot.ijccce.22.4.9

Malallah, F. L., Abduljabbar, A. I., Shareef, B. T., & Al-Janaby, A. O. (2023). QR Code Encryption for improving Bank information and Confidentiality. *2023 27th International Conference on Information Technology (IT)*. https://doi.org/10.1109/IT57431.2023.10078457

Miftahul Amri, M., Waeno, M., & Zain Musa, M. (2023). LSB Steganography to Embed Creator's Watermark in Batik Digital Arts. *Engineering Science Letter*, *2*(01), 27–32. https://doi.org/10.56741/esl.v2i01.301

Mouhsen, S. B., & Hussain, Z. M. (2022). Watermarking Using Energy-LSB Embedded Method. *Wasit Journal of Computer and Mathematics Science*, *1*(3), 89–94. https://doi.org/10.31185/wjcm.53

Mrs. M. Saritha, Ranjith. S, Vishvanth. R, & Vijay. R. (2023). VLSI Implementation of AES Algorithm in Cryptography Developed in Xilinx. *International Journal of Advanced Research in Science, Communication and Technology*, 558–562. https://doi.org/10.48175/ijarsct-9617

numpy · PyPI. (2023). https://pypi.org/project/numpy/

Pan, J. S., Sun, X. X., Chu, S. C., Abraham, A., & Yan, B. (2021). Digital watermarking with improved SMS applied for QR code. *Engineering Applications of Artificial Intelligence*, *97*, 104049. https://doi.org/10.1016/J.ENGAPPAI.2020.104049

Picard, J., Landry, P., & Bolay, M. (2021). Counterfeit detection with QR codes. *DocEng '21: Proceedings of the 21st ACM Symposium on Document Engineering*, 1–4. https://doi.org/10.1145/3469096.3474924

Pillow (PIL Fork) 10.4.0 documentation. (2023). https://pillow.readthedocs.io/en/stable/

Priowirjanto, E. S., Suparman, E., Amirulloh, M., & Rahmawati, E. (2022). QR Codes to Prevent Copyright Infringement: Case Study of Trusmi Batik in Cirebon, Indonesia. *Journal of Applied Security Research*, *19*(2), 161–167. https://doi.org/10.1080/19361610.2022.2113731

Purdadi, I. G., Al Anshori, F. A., & Alfitrah, M. D. (2023). Implementasi Teknologi QR Code Pada Pengarsipan Bukti Pembayaran di kampus IIB Darmajaya. *Journal of Digital Literacy and Volunteering*, *1*(1), 34–40. https://doi.org/10.57119/ict.v1i1.18

pycryptodome · PyPI. (2024). https://pypi.org/project/pycryptodome/

qrcode · PyPI. (2023). https://pypi.org/project/qrcode/

Rajeswari, C., Babu, S., & Venkatesan, P. (2013). Analysis of MPC Image Compression using DCT 2 in Matlab. *International Journal of Computer Applications*, *73*(14), 25–30. https://doi.org/10.5120/12809-0050

Rani, D. N. U., Anjum, S. D., Vishal, K., Reddy, N. H., & Akram, S. S. (2023). Repository and Retrieval of Data using AES Security in Cloud Computing Environment. *International Journal for Research in Applied Science and Engineering Technology*, *11*(4), 4224–4229. https://doi.org/10.22214/ijraset.2023.51255

S, K., S Adithya, K., Franklin A, I., Prasath S, H., & M, L. (2023). Advanced Encryption Standard to Prevent Intruders in Email through Cloud Environment. *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*. https://doi.org/10.1109/ICAAIC56838.2023.10140373

Sakshi, S., Verma, S., Chaturvedi, P., & Yadav, S. A. (2022). Least Significant Bit Steganography for Text and Image hiding. *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*. https://doi.org/10.1109/ICIEM54221.2022.9853052

Shete, P., & Kohle, S. (2022). Image Encryption using AES Algorithm: Study and Evaluation. *International Journal for Research in Applied Science and Engineering Technology*, *10*(9), 1134–1137. https://doi.org/10.22214/ijraset.2022.46619

Shokeen, G., Aggarwal, S., & Bhatia, D. M. K. (2022). QR Code Analysis. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, *10*(12), 1–23.

Taufiqqurrachman, T., & Elsandi, D. (2022). Security Analysis and Encryption Time Comparison Description on Cryptography Advanced Encryption Standard (AES). *Jurnal Inovatif : Inovasi Teknologi Informasi Dan Informatika*, *5*(1), 60. https://doi.org/10.32832/inovatif.v5i1.8345

Teguig, E. ., Touati, Y., & Ali-Cherif, A. (2017). ECC Based-Approach for Keys Authentication and Security in WSN. *2017 9th IEEE-GCC Conference and Exhibition (GCCCE)*. https://doi.org/10.1109/IEEEGCC.2017.8447901

Tsai, M.-J., & Hsieh, C.-Y. (2019). The visual color QR code algorithm (DWT-QR) based on wavelet transform and human vision system. *Multimedia Tools and Applications*, *78*, 21423–21454. https://doi.org/10.1007/s11042-019-7308-y

Tsai, M. J., Lee, Y. C., & Chen, T. M. (2023). Implementing Deep Convolutional Neural Networks for QR Code-Based Printed Source Identification. *Algorithms*, *16*(3). https://doi.org/10.3390/a16030160

Wade, S. (2023). Description of Image encryption Using AES-256 bits. *International Journal for Research in Applied Science and Engineering Technology*, *11*(5), 7167–7171. https://doi.org/10.22214/ijraset.2023.53365

Wahsheh, H. A. ., & S., A. M. (2022). QR Codes Cryptography: A Lightweight Paradigm. *International Conference on Information Systems and Intelligent Applications (ICISIA 2022)*, 649–658. https://doi.org/10.1007/978-3-031-16865-9_52

Wave, D. (2024). *QR Code development story*. https://www.denso-wave.com/en/technology/vol1.html

Widiyono, W., Wibowo Putra, A., Risqiati, R., & Syaifudin, A. (2022). Perlindungan Data Informasi Digital Dengan Teknik Steganografi Metode Least Significant Bit. *Smart Comp: Jurnalnya Orang Pintar Komputer*, *11*(3), 323–331. https://doi.org/10.30591/smartcomp.v11i3.3453

Wu, W., Zhang, L., Zhang, J., Cui, C., Zhang, X., & Liu, M. (2022). A three-level QR code sharing scheme based on SLIC and Hamming code. *2022 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. https://doi.org/10.1109/ICSPCC55723.2022.9984448

Xuan, X., Zhang, X., Kwon, O. H., & Ma, K. L. (2022). VAC-CNN: A Visual Analytics System for Comparative Studies of Deep Convolutional Neural Networks. *IEEE Transactions on Visualization and Computer Graphics*, *28*(6), 2326–2337. https://doi.org/10.1109/TVCG.2022.3165347

Yao, Y., Wang, L., & Shen, J. (2022). Features and Applications of QR Codes. *International Journal for Innovation Education and Research*, *10*(5), 166–169. https://doi.org/10.31686/ijier.vol10.iss5.3762

Zhang, L. N., Cui, C. Y., Zhang, X. Y., & Wu, W. (2022). Adaptive visual cryptography scheme design based on QR codes. *Mathematical Biosciences and Engineering*, *19*(12), 12160–12179. https://doi.org/10.3934/mbe.2022566