REICE
Revista Electrónica de Investigación en Ciencias Económicas
Abriendo Camino al Conocimiento
Área de Conocimiento de Ciencias Económicas y Administrativas
Universidad Nacional Autónoma de Nicaragua, Managua (UNAN-Managua)

Vol. 12, No. 24, julio – diciembre 2024 **REICE ISSN: 2308-782X**
https://revistas.unan.edu.ni/index.php/reice
revista.reice@unan.edu.ni

Method for Digitally Signing Document Images for Their Verification in Electronic Document Management Systems

Método para firmar digitalmente imágenes de documentos para su verificación en sistemas de gestión de documentos electrónicos

Fecha de recepción: julio 05 de 2024
Fecha de aceptación: septiembre 12 de 2024
DOI: https://doi.org/10.5377/reice.v12i24.20087

Elena Avksentieva
ITMO University, Saint Petersburg, Russia
Email: avksentievaelena@rambler.ru
ORCID: https://orcid.org/0000-0001-5000-4868

Evgenii Mazayshvili
ITMO University, Saint Petersburg, Russia
Email: evgenij997@yandex.ru
ORCID: https://orcid.org/0000-0001-8592-0751

Abstract

The article discusses a method for digitally signing document images without using metadata or additional files. The method is resistant to jpeg compression within certain limits. The object of the digital signature is 8x8 pixel blocks with a discrete cosine transform applied, which make up the jpeg file. A method to achieve resistance to jpeg compression is to quantize (reduce the precision) the values in the blocks so that when the signature is calculated, they remain unchanged up to a certain compression level. The study presents a robust method for digitally signing JPEG images that remains effective despite compression. This method is particularly useful for verifying the authenticity of document images in electronic document management systems without requiring extensive infrastructure or certification processes.

***Keywords*:** Digital Signature, JPEG, PNG, Discrete Cosine Transform.

Resumen

El artículo analiza un método para firmar digitalmente imágenes de documentos sin utilizar metadatos ni archivos adicionales. El método es resistente a la compresión jpeg dentro de ciertos límites. El objeto de la firma digital son bloques de 8x8 píxeles a los que se les aplica una transformación de coseno discreto, que conforman el archivo jpeg. Un método para lograr resistencia a la compresión jpeg es cuantizar (reducir la precisión) los valores en los bloques para que cuando se calcule la firma, permanezcan sin cambios hasta un cierto nivel de compresión. El estudio presenta un método sólido para firmar digitalmente imágenes JPEG que sigue siendo eficaz a pesar de la compresión. Este método es particularmente útil para verificar la autenticidad de las imágenes de documentos en sistemas de gestión de documentos electrónicos sin requerir una infraestructura extensa o procesos de certificación.

Palabras claves: Firma digital, JPEG, PNG, Transformada de coseno discreta.

Introduction

An electronic digital signature (EDS), according to the law, is a complete analogue of a handwritten signature, therefore files and reports certified by it have full legal force (Grudtsina et al. 2022). To prevent fraud and accidental errors when generating digital signatures, you must submit a set of documents to an accredited certification center (Telnov et al. 2024). It serves to fully identify the future owner of the signature and confirm the rights to own the token (Gladilina et al. 2023). The specific list of papers depends on the type of applicant: an individual or an individual entrepreneur, a legal entity (Yessenali et al. 2024).

Having your own electronic digital signature will allow you to remotely contact government agencies and correspond with them (Levin et al. 2023). With its help, you can submit applications for admission to educational institutions without personally visiting them, sign an agreement with an employer without visiting the latter's office, and perform other actions (Abdullaev et al. 2023). As a result, having a token with a signature saves time on visits to government agencies by receiving services in a remote format (Vinichenko et al. 2021).

However, there is often a need to quickly certify a large volume of documents that do not require a qualified electronic signature according to the law (Rybak et al. 2023), but their authenticity is very important for the organization (Mirzabalaeva et al. 2019), for example, images of medical certificates of employees or students, which are collected by the employer or university via transmission over the Internet (Gurinovich & Petrykina, 2021), in particular, through instant messengers and social networks, as well as to confirm the authenticity of documents for electronic document management systems (Ketova & Ovchinnikov, 2024).

A digital signature is a mathematical scheme that allows you to verify the authenticity and integrity of a document presented in digital form (Bellare & Goldwasser, 2008). A modern cryptographic digital signature algorithm can be applied to any digital file (Wu & Nachiangmai, 2024). Most implementations involve creating an additional file in such a way that the original file + digital signature file is considered the signed document (Lolaeva et al. 2022). Many file formats, such as PDF, support embedding a digital signature within the source file (International Organization for Standardization, 2008). In this case, the original file with an embedded digital signature will be considered a signed document.

However, for digital images in the two most popular formats, JPEG and PNG, there is no scheme for storing a digital signature within the source file (Kirillova et al. 2021). Thus, there are only two options for implementing a digital signature for image:

1. Embedding a signature into the image metadata, for example, dSIG chunks for PNG images and EXIF metadata for JPEG images (Ermakov et al. 2022).

2. Saving a separate digital signature file in addition to the original file.

Both implementations have the disadvantage of the inability to transfer signeddocument images through uploading to popular social networks, instant messengers, popular image hostsites and electronic document management systems (Safronova et al. 2023; Kenzhin et al. 2021). The reason is that they perform transcoding of images, removing metadata. Many of them perform jpeg compression in addition to removing metadata.In this regard, it became necessary to create a digital signature of the document image that can be verified after transmitting the image through instant messengers and social networks (after jpeg compression and removal of all metadata) (Abdullaev et al. 2023).

The goal of this work is to create a method for embedding a digital signature into a JPEG image by modifying the image itself - adding additional pixels containing the signature to its right side. The signature must be resistant to jpeg compression within certain limits and used for black and white images.

Methodologies and Data

Proposed method for digital signature of document images

The main problem when implementing a digital signature for signing images that is resistant to compression is the lack of a guarantee that after compression the colors of the image pixels will remain unchanged. Therefore, an array of all pixels cannot be used as a signature object. Also, approximate pixel values cannot be used as a digital signature object. In this case, an attacker has the ability to change the colors of pixels within certain limits, forming an incorrect image, without violating the signature.

The proposed document image signature method consists of six steps

1. Calculation of an array of DCT blocks from a black-and-white source image.

2. Quantization of DCT blocks to a given value.

3. Calculation of the parity table for each DCT block. This table will allow the digital signature to remain valid with minor changes in the DCT block.

4. Digitallysigning an array of DCT blocks using the RSA algorithm.

5. Encoding the resulting signature and parity tables into a separate image (hereinafter referred to as: image-signature, see Figure 7, right) using a method that gives the image-signature resistance to jpeg compression.

6. Attaching the image-signature to the initial image.

The document image signature verification method also consists of six steps:

1. Dividing the signed image into the initial image and the image-signature.

2. Decryption of data from the image-signature - obtaining a signature in a form ready for verification and parity tables.

3. Calculation of an array of DCT blocks from the initial image.

4. Quantization of DCT blocks to a given value

5. Making amendments to DCT blocks from parity tables.

6. Check: whether the digital signature read in step 2 matches the hash of the DCT block array.

The method has some similarities with steganography, based on hiding data in DCT coefficients (Awad Attaby et al., 2018; Buchanan, 2024). The coefficients are analyzed, and the result of this analysis reveals initially hidden information - whether the signature is valid or not. However, this method is not intended to hide information inside jpeg images. When looking at the final image, you will see that it consists of two parts - the image itself and its signature.

It is important to note that the digital signature must itself be an image and be attached to the image being signed. This will make it susceptible to jpeg compression, which will break the signature. Therefore, the signature must be consistently readable when the image is compressed within certain limits.

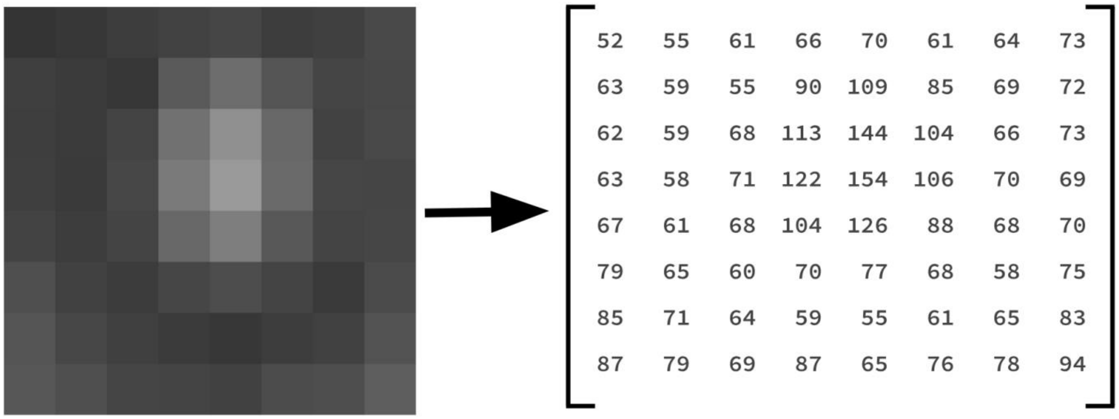
Results and discussion

Creation of digital signature in accordance with the proposed method

The initial image can be either a lossless compressed image such as png or a loss compressed image such as jpeg. The process for lossless compressed image will be described below. The process of compressing a jpeg file differs only in the absence of the need to calculate the DCT-block array, as it is already contained in the file.

Step 1: Calculating the array of DCT-blocks from the initial image.

Discrete Cosine Transform (DCT) is an algorithm that transforms an image from pixel space to frequency space, where each 8x8 pixel block corresponds to an 8x8 block of coefficients (Griffin, 2023). An example of a DCT block is shown in Figure 1.



$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 87 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Figure 1. Discrete cosine transform of an 8x8 pixel plot

The values in the DCT block have a range from -1023 to 1023 in steps of 1. A larger value corresponds to a larger amplitude of the corresponding cosine function and a larger brightness of certain pixels depending on the function.

Excluding rounding errors, DCT is a lossless transformation algorithm - the table can be converted back to the original pixel array.

In the first step of the method, the initial image is converted into an array of DCT blocks.

Step 2: Quantization of DCT-blocks by a given value

The second step makes the values in the DCT-blocks coarser by performing quantizationwith a specific step. Quantization is a transformation where each value within a DCT-block is transformed using formula (1):

$$value'_{x,y} = floor\left(\frac{value_{x,y}}{Q_{x,y}}\right) \times Q_{x,y} \qquad (1)$$

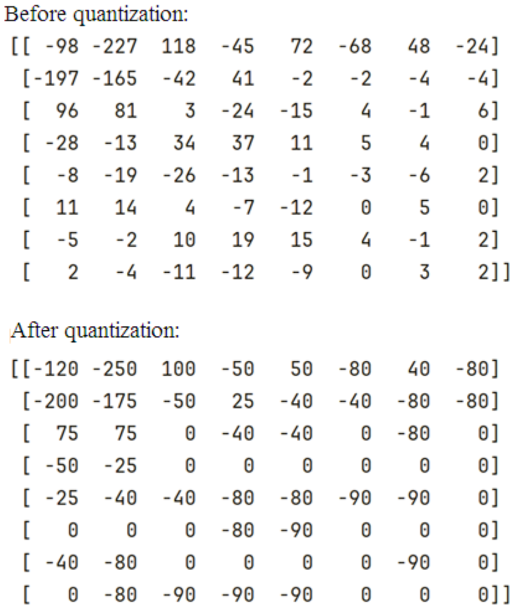
where $Q_{x,y}$ is the value of quantization step from the table of quantizationsteps.

The method uses a variable step value depending on the location of the value in the table: the step increases towards the lower right corner of the table. The reason for this is that the jpeg compression algorithm introduces more distortion to values closer to the lower right corner of the DCT block, where finer details of the image are represented. Since we want to preserve more explicit details more accurately, smaller quantization steps are used for values closer to the upper left corner of the table. The value in the upper left corner has a larger step because changing it only changes the overall brightness of the block in the final image. The used step values are marked in Figure 2.

$$\begin{bmatrix} 40 & 25 & 25 & 25 & 25 & 40 & 40 & 80 \\ 25 & 25 & 25 & 25 & 40 & 40 & 80 & 80 \\ 25 & 25 & 25 & 40 & 40 & 80 & 80 & 90 \\ 25 & 25 & 40 & 40 & 80 & 80 & 90 & 90 \\ 25 & 40 & 40 & 80 & 80 & 90 & 90 & 90 \\ 40 & 40 & 80 & 80 & 90 & 90 & 90 & 90 \\ 40 & 80 & 80 & 90 & 90 & 90 & 90 & 90 \\ 80 & 80 & 90 & 90 & 90 & 90 & 90 & 90 \end{bmatrix}$$

Figure 2. Value of quantization step depending on the place of the value in the block

The point of quantization can be thought of as moving away from the exact values in the block to working with ranges of values. For example, a value of 529 at 60% compression will be distorted to 521, but, with a quantization step of 10, they will both fall within the range 520-530. Similarly, with a quantization step of 50, both values will fall in the range 500-550. The lower value of the range is exactly the result of quantization with a particular step, as in Figure 3.



```
Before quantization:
[[ -98 -227  118  -45   72  -68   48  -24]
 [-197 -165  -42   41   -2   -2   -4   -4]
 [  96   81    3  -24  -15    4   -1    6]
 [ -28  -13   34   37   11    5    4    0]
 [  -8  -19  -26  -13   -1   -3   -6    2]
 [  11   14    4   -7  -12    0    5    0]
 [  -5   -2   10   19   15    4   -1    2]
 [   2   -4  -11  -12   -9    0    3    2]]

After quantization:
[[-120 -250  100  -50   50  -80   40  -80]
 [-200 -175  -50   25  -40  -40  -80  -80]
 [  75   75    0  -40  -40    0  -80    0]
 [ -50  -25    0    0    0    0    0    0]
 [ -25  -40  -40  -80  -80  -90  -90    0]
 [   0    0    0  -80  -90    0    0    0]
 [ -40  -80    0    0    0    0  -90    0]
 [   0  -80  -90  -90  -90    0    0    0]]
```

Figure 3. Values in one of the DCT tables before and after quantization (quantization steps in Figure 2 are used)

The quantized blocks are not converted back to pixels and are not stored in the final signed image. They are needed only for the step of calculating their hash. It is the hash of all quantized blocks that is the object of the digital signature.

If the object of signature was approximate pixel values, instead of approximate DCT values, an attacker could change them within approximation, drawing the adversary details in the image. This work uses approximate values, not of pixels, but of DCT blocks, changes of which by an attacker will only lead to changes in subtle details in the image.

Step 3. Calculate the parity table for each DCT block.

When quantizing a block, it may be that some values are at or close to the boundary of the quantization interval. For example, for quantization with a step of 10, the value 50 will be on the boundary of the intervals 40-50 and 50-60. In this case, even a small distortion during compression can shift the value into the wrong interval, making the signature invalid.

In order to avoid this problem, each of the intervals determined by the quantization step is assigned a parity, as in Figure 4.

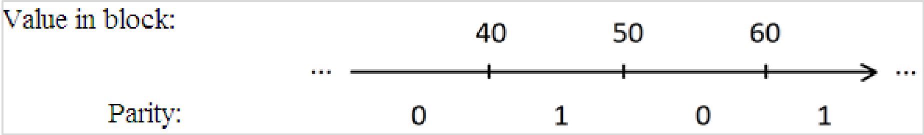
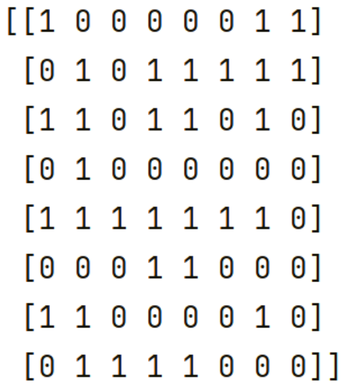


Figure 4. Parity for each value in the quantized block

For example, with a quantization step of 10, the value 51 is in the odd interval 50-60, and the value 49 is in the even interval 40-50.Together with the digital signature, the parity of each value for each block is saved. It is stored as a single bit.

Thus, for each quantized block the corresponding parity table is calculated. This table allows you to make corrections to the block values at the decryption stage if the value during compression accidentally crosses the interval boundary. An example of a parity table in Figure 5.



```
[[1 0 0 0 0 0 1 1]
 [0 1 0 1 1 1 1 1]
 [1 1 0 1 1 0 1 0]
 [0 1 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 0]
 [0 0 0 1 1 0 0 0]
 [1 1 0 0 0 0 1 0]
 [0 1 1 1 1 0 0 0]]
```

Figure 5. Parity table for the values from the table below in Figure 3.

The table occupies 64 bits for each 8x8 pixel image area.

Step 4. Digitally signing the hash of an array of DCT blocks using the RSA algorithm

The array of all quantized blocks and the array of parity tables are hashed using the SHA-256 algorithm and the resulting hash is signed with a private key according to the RSA algorithm.

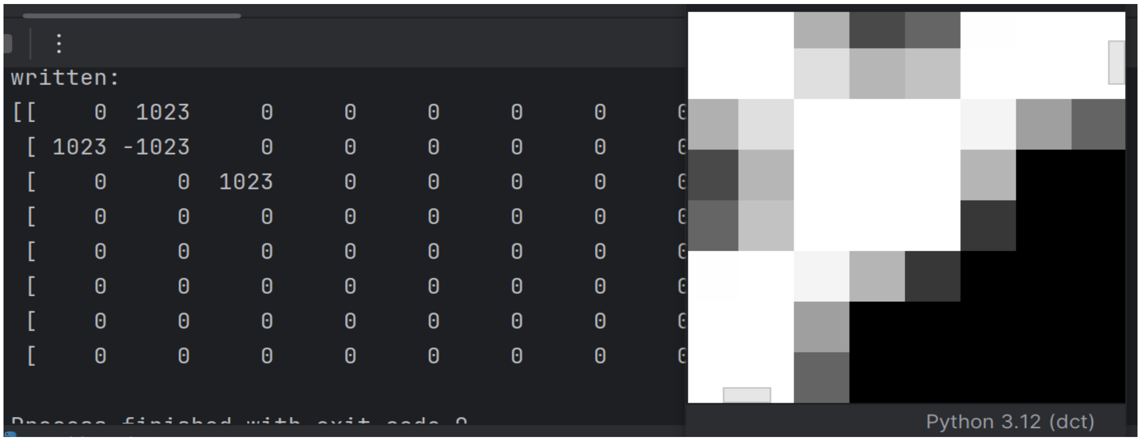
As a result of the fourth step, only two artifacts are obtained that are necessary for signature verification: a signed hash of 2048 bits in size (using RSA-2048) and an array of parity tables with size of 64 bits times number of 8x8 pixel blocks in the image.

Step 5. Encoding the received signature and parity tables into a image-signature

The signed hash and parity tables must be stored directly in the image, and to do this they must be resistant to jpeg compression.

In order to achieve compression resistance, one of the features of the operation of the discrete cosine transform was investigated. A DCT block is an 8x8 block of values, from -1023 to 1023. A block consisting of only maximum and minimum values will exhibit resistance to compression, since compression of such a block will result in only minor distortions being introduced into the values. For example, a value of -1023 might become a value of -990 after compression. However, its proximity to the minimum value will allow us to unambiguously consider it the minimum value. Thus, an image composed of DCT blocks with a maximum and minimum value will be resistant to very high compression level.

In this work, three values are used: -1023, 0 and 1023. Thus, in one value of the DCT block in the signature image, 2.5 bits of useful information are stored. An example of such a block is shown in Figure 6.



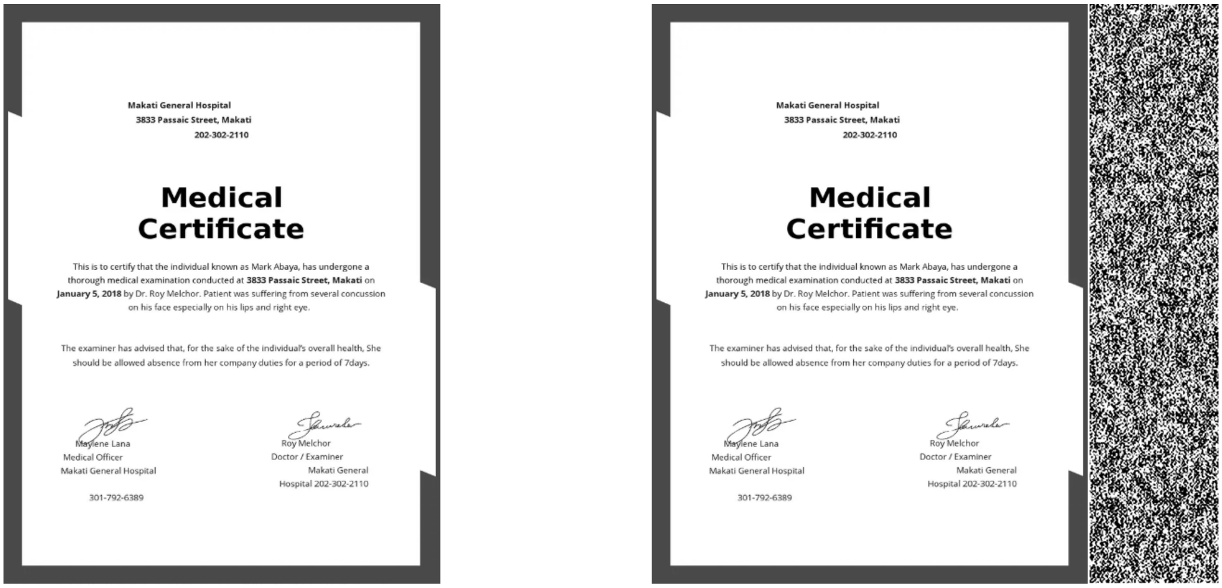
```
written:
[[    0  1023     0     0     0     0     0     0
 [ 1023 -1023     0     0     0     0     0     0
 [    0     0  1023     0     0     0     0     0
 [    0     0     0     0     0     0     0     0
 [    0     0     0     0     0     0     0     0
 [    0     0     0     0     0     0     0     0
 [    0     0     0     0     0     0     0     0
 [    0     0     0     0     0     0     0     0

Process finished with exit code 0
```

Figure 6. An artificially created DCT block and its corresponding block of pixels, which will be included in the final image-signature

Thus, encoding 64^2.5 bits into an 8x8 pixel image, all parity tables will fit into an area occupying no more than 1/3 of the image.

Step 6: Attaching the image-signature to the initial image

After encoding, the signature image containing the signed hash and parity tables is attached to the initial image (see Figure 7).



Makati General Hospital
3833 Passaic Street, Makati
202-302-2110

Medical Certificate

This is to certify that the individual known as Mark Abaya, has undergone a thorough medical examination conducted at **3833 Passaic Street, Makati** on **January 5, 2018** by Dr. Roy Melchor. Patient was suffering from several concussion on his face especially on his lips and right eye.

The examiner has advised that, for the sake of the individual's overall health, She should be allowed absence from her company duties for a period of 7days.

Maylene Lana
Medical Officer
Makati General Hospital

301-792-6389

Roy Melchor
Doctor / Examiner
Makati General
Hospital 202-302-2110

Makati General Hospital
3833 Passaic Street, Makati
202-302-2110

Medical Certificate

This is to certify that the individual known as Mark Abaya, has undergone a thorough medical examination conducted at **3833 Passaic Street, Makati** on **January 5, 2018** by Dr. Roy Melchor. Patient was suffering from several concussion on his face especially on his lips and right eye.

The examiner has advised that, for the sake of the individual's overall health, She should be allowed absence from her company duties for a period of 7days.

Maylene Lana
Medical Officer
Makati General Hospital

301-792-6389

Roy Melchor
Doctor / Examiner
Makati General
Hospital 202-302-2110

Figure 7. From left to right: initial image, digitally signed image

The size of the signed area is always a fixed part of the image, because the number of parity tables linearly depends on the image size, and the signed hash has a fixed length. This step completes the signing process.

Verification of Digital Signature

The input data for the verification step is a digitally signed image compressed by the jpeg algorithm.

Step 1: Separation of the signed image into initial image and image-signature

Since the size of the digital signature is known, dividing the images into initial image and image-signature is trivial.

Step 2: Decoding data from the signature image - obtaining the signature ready for verification and parity tables.

To decrypt the signature image, each 8x8 pixel segment is converted into a DCT-block, the values of which are rounded to the nearest - "1023", "0" or "-1023".

Thus, the signed hash and parity tables are restored.

Step 3: Computing an array of DCT-blocks from the initial image.

The step is similar to the first step in signing the image.

DCT calculation is not required if the file is already in jpeg format, in this case it is enough to read ready DCT-blocks from the file.

Step 4: Quantization of DCT-blocks by specified value

The quantization step is completely the same as the one for image signing. The purpose of this step is to get the same quantized blocks that were in the image at the time of signing so that the signature can be validated.

Step 5: Correcting DCT blocks from parity tables

At the fifth step the parity tables for quantized blocks are calculated, similarly to the signature step. However, in contrast to the similar step of signing, during verification the tables are compared against the real values in blocks, and based on the results of the comparison the values of blocks are corrected: those values, which may have exceeded the boundary of their interval, because they were close to it, are returned to their interval.

If the calculated parity of the value in the block does not coincide with the corresponding parity recorded in the parity table in the signature, correction is performed.

The algorithm of this correction is described by formula (2):

$$v'_{x,y} = \begin{cases} \left|\overline{v}_{x,y} - v_{x,y}\right| > \frac{Q_{x,y}}{2} : v_{x,y} + \frac{Q_{x,y}}{2} \\ else : v_{x,y} - \frac{Q_{x,y}}{2} \end{cases} \qquad (2)$$

where $Q_{x,y}$ is the table of quantization steps, $v_{x,y}$is thequantized value in the block, $\bar{v}_{x,y}$ is the value in the block before quantization.

The formula describes the following transformation. If a value is closer to the lower boundary of an interval, it is considered to have entered the interval by mistake from a lower interval. The interval in the quantized table is changed to a lower interval. If the value is closer to the upper boundary of the interval, then it is considered to have entered it by mistake from the higher interval. The interval in the quantized table is changed to the higher one.

Step 6: Checking whether the digital signature read in step 2 matches the calculated hash.

The last step is to hash the array of DCT tables and parity tables from the signature using the same scheme as in image signing, after which the resulting hash is compared with the one in the digital signature. A matching hash indicates that the signature is correct. A mismatch indicates either too much compression or deliberate distortion of the image.

Conclusion

In this paper, a method for digitally signing images in JPEG format as well as in any lossless compression format such as PNG has been proposed. The method is robust to jpeg compression by using quantized DCT blocks as the signature object. The compression robustness is achieved by storing the digital signature data as an image consisting of DCTmaximum amplitude basis functions.

The method can be used as a simplified scheme for verification of images of documents transmitted via the Internet, in particular, via messengers or social networks, which does not require, according to the legislation, application to accredited certification centers.

As a perspective for the development of the method, it is possible to increase the data density in the signature image so that it occupies a smaller part of the main image, which is important for increasing the speed of file transfer on the Internet and the operation of electronic document management systems.

References

Abdullaev, I., Prodanova, N., Ahmed, M. A., Joshi, G. P., & Cho, W. (2023). Leveraging metaheuristics with artificial intelligence for customer churn prediction in telecom industries. Electronic Research Archive, 31(8), 4443-4458. https://doi.org/10.3934/era.2023227

Abdullaev, I., Prodanova, N., Bhaskar, K. A., Lydia, E. L., Kadry, S., Kim, J. (2023). Task offloading and resource allocation in iot based mobile edge computing using deep learning. Computers, Materials & Continua, 76(2), 1463-1477. doi: 10.32604/cmc.2023.038417

Awad Attaby, A., Mursi Ahmed, M. F. M., & Alsammak, A. K. (2018). Data hiding inside JPEG images with high resistance to steganalysis using a novel technique: DCT-M3. Ain Shams Engineering Journal, 9(4), 1965-1974. http://dx.doi.org/10.1016/j.asej.2017.02.003

Bellare, M., & Goldwasser, S. (2008). Digital signatures. In Lecture notes on cryptography (pp. 168-205). Cambridge, Massachusetts. Retrieved from https://cseweb.ucsd.edu//~mihir/papers/gb.pdf

Buchanan, W. J (2024). DCT (Discrete Cosine Transform). Retrieved from https://asecuritysite.com/comms/dct2

Ermakov, S., Pcholovsky, N., Vasyukov, V., Rodkina, N., & Mikhaylenko, N. (2022). Illegal use of foreign trademarks in the Russian Federation: Issues of qualification and investigation. Lex Humana, 14(2), 231–244. Retrieved from https://seer.ucp.br/seer/index.php/LexHumana/article/view/2292

Gladilina, I. ., Degtev, G., Kochetkov, E., Tretyak, E., Stepanova, D., & Mutaliyeva, L. (2023). Development of User Subscription Services in E-Commerce: Effects on Consumer Behavior. Revista Electrónica De Investigación En Ciencias Económicas, 10(20), 53–67. https://doi.org/10.5377/reice.v10i20.16026

Griffin, J. (2023, January 4). The ultimate guide to JPEG including JPEG compression & encoding. Retrieved from https://www.thewebmaster.com/jpeg-definitive-guide/

Grudtsina, L., Guliyeva, M. E. K., Zhdanov, S., Sangadzhiev, B., & Shestak, V. (2022). Application of digital technologies in law. Jurnal Cita Hukum - Indonesian Law Journal, 10(3).

Gurinovich, A. G., & Petrykina, N. I. (2021). Características del desarrollo de la Institución de Servicio Público: Experiencia Internacional y su aplicación en Rusia. JURÍDICAS CUC, 17(1), 253–276. https://doi.org/10.17981/juridcuc.17.1.2021.09

International Organization for Standardization. (2008). ISO 32000-1:2008. Document management - Portable document formatPart 1: PDF 1.7. Retrieved from https://www.iso.org/standard/51502.html

Kenzhin, Z. B., Tulegenova, A. U., Zolkin, A. L., Kosnikova, O. V., & Shichkin, I. A. (2021). Labour market under economy digitalization. E3S Web of Conferences, 311, 08007. https://doi.org/10.1051/e3sconf/202131108007

Ketova, N., & Ovchinnikov, V. (2024). Strategic Management Of Innovative Changes In The Russian Economy: Assessments And Main Approaches. Revista Gestão & Tecnologia, 24(2), 258–269. https://doi.org/10.20397/2177-6652/2024.v24i2.2763

Kirillova, E. A., Zulfugarzade, T. E., Blinkov, O. E., Serova, O. A., & Mikhaylova, I. A. (2021). Perspectivas de desarrollo de la regulación legal de las plataformas digitales. JURÍDICAS CUC, 18(1), 35–52. https://doi.org/10.17981/juridcuc.18.1.2022.02

Levin, M., Novikova, M., & Filatova, I. . (2023). Impact of Global Threats on Economic Security. Revista Electrónica De Investigación En Ciencias Económicas, 10(20), 43–52. https://doi.org/10.5377/reice.v10i20.16025

Lolaeva, A., Lebedeva, M., Matveeva, N., Nesmeianova, I., Ocheredko, V., & Platonova, S. (2022). Digital (electronic) democracy in Russia: Issues of further development. Jurnal Cita Hukum - Indonesian Law Journal, 10(3).

Mirzabalaeva, F. I., Shichkin, I. A., & Neterebsky, O. V. (2019). Economic depression in regional labor markets and subsidy dependence of regions. International Journal of Civil Engineering and Technology, 10(2), 1838–1845.

Rybak, V., Kryanev, Y., Shichkin, I., & Livson, M. (2023). State regulation as a comprehensive mechanism for the sustainable development of territories. Revista Juridica, 1(73), 831-844. http://dx.doi.org/10.26668/revistajur.2316-753X.v1i73.6282

Safronova, N., Nezhnikova, E., & Papelniuk, O. (2023). Assessment of the impact of satisfaction with management companies on housing and communal services market development. Quality - Access to Success, 24(193), 230

Telnov, Y., Kazakov, V., Danilov, A., & Fiodorov, I. (2024). Network enterprise architecture based on multiagent technology. Revista Gestão & Tecnologia, 24(2), 66–95. https://doi.org/10.20397/2177-6652/2024.v24i1.2719

Vinichenko, M. V., Klementyev, D. S., Rybakova, M. V., Malyshev, M. A., & Malysheva, N. S. (2021). Satisfaction with the quality of life in employees of Russian enterprises in the social partnership system. Quality - Access to Success, 22(180)

Wu, J., & Nachiangmai, S. (2024). Relationship between open innovation and innovation performance within high-tech firms: The mediating role of knowledge management capability. Journal of Infrastructure, Policy and Development, 8(5), 3887. doi:http://dx.doi.org/10.24294/jipd.v8i5.3887

Yessenali, A., Azretbergenova, G., Izatullayeva, B., Baibosynova, G., Zhetibayev, Z., Myrzaliev, B., Biryukov, V., & Shichiyakh, R. (2024). Conceptual platform and applied aspects of interaction between government and business agencies in the Republic of Kazakhstan. Journal of Infrastructure, Policy and Development, 8(5), 3396. doi:http://dx.doi.org/10.24294/jipd.v8i5.3396.