International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 4, August 2023, pp. 4388~4400
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i4.pp4388-4400

Approximation-based homomorphic encryption for secure and efficient blockchain-driven watermarking service

Didi Rosiyadi[1], Akbari Indra Basuki[1], Taufik Iqbal Ramdhani[1], Heru Susanto[1,2,3], Yusnan Hasani Siregar[4]
[1]Research Center for Data and Information Sciences, National Research and Innovation Agency, Bandung, Indonesia
[2]Information Management Department, Tunghai University, Taichung, Taiwan
[3]Center for Innovative Engineering, University of Technology Brunei, Bandar Seri Begawan, Brunei Darussalam
[4]Research Center for Appropriate Technology, National Research and Innovation Agency, Subang, Indonesia

Article Info

Article history:

Received Jul 15, 2022
Revised Nov 2, 2022
Accepted Nov 6, 2022

Keywords:

Blockchain
Cheon-Kim-Kim-Son homomorphic encryption
Discrete cosine transform-singular value decomposition
Encrypted watermarking
Watermarking service

ABSTRACT

Homomorphic encryption has been widely used to preserve the privacy of watermarking process on blockchain-driven watermarking services. It offers transparent and traceable encrypted watermarking without revealing sensitive data such as original images or watermark data to the public. Nevertheless, the existing works suffer from enormous memory storage and extensive computing power. This study proposed an approximation-based homomorphic encryption for resource-efficient encrypted watermarking without sacrificing watermarking quality. We demonstrated the efficiency of the Cheon-Kim-Kim-Son (CKKS) encrypted watermarking process using discrete cosine transform-singular value decomposition (DCT-SVD) embedding. The evaluation results showed that it could preserve the watermarking quality similar to non-encrypted watermark embedding, even after geometrical and filtering attacks. Compared to existing homomorphic encryption, such as Brakerski-Gentry-Vaikuntanathan (BFV) encryption, it has superior performance regarding resource utilization and watermarking quality preservation.

Corresponding Author:

Didi Rosiyadi
Research Center for Data and Information Sciences, National Research and Innovation Agency
Cisitu Street, No. 21 Bandung, West Java, Indonesia
Email: didi.rosiyadi@brin.go.id

1. INTRODUCTION

Blockchain-based watermarking service offers better transparency and data integrity than conventional web or cloud watermarking services. In a web-based model, the entire business process of watermarking services is opaque to end users [1]–[3]. In addition, the system administrator has full access to the system, allowing them to abuse the watermarking services. Untrusted administrators might issue valid watermarked images without the consent of the content creator or reject the watermarking request one-sidedly.

Several works proposed secure and transparent methods for web-based watermarking protocols, either by using trusted authority [4]–[6] or client-side embedding [7]–[10]. Despite the efforts, the works cannot provide full transparency for the watermarking process. Authority-based watermarking has a weakness in the authority's integrity. The security will be in disarray if the authority colludes with the buyer or content provider to sell a duplicate of the watermarked image. Client-side embedding, however, suffers from dispute resolution between buyer and seller. A trusted authority is the core component of settling any conflicts between the buyer and seller. Thus, the main challenge is how to ensure the authority's integrity in a transparent manner.

Some studies use blockchain to improve the transparency of the watermarking service. Blockchain offers transparent and immutable transactions that enable public monitoring of a business process.

Unfortunately, the existing works [11]–[15] limit the work for transaction recordings and do not address the transparency of the watermarking process. In this case, the content provider one-sidedly runs the watermarking process while the authority's role is to maintain a secure communication system between buyer and seller. Based on our survey, only the works in [16], [17] offer fully transparent watermark embedding based on blockchain. The works share the homomorphic-encrypted image data in the blockchain for public monitoring. Anyone can prove the correctness of watermark embedding by recomputing the embedding function. Meanwhile, only the buyer can decrypt the result since the data are encrypted using the buyer's public key.

Despite its optimal combination, blockchain and homomorphic encryption scheme suffer in practicability. Computing the homomorphic-encrypted watermarking requires vast memory space and enormous computation time, particularly for large image sizes. Basuki *et al*. [17] mitigated this problem by using Hadamard product decomposition to reduce memory usage. It breaks down the matrix operation into per-row multiplication to reduce the memory demand into a linear scale. Despite its support for parallel computation, the work fails to ensure computation tractability since it fails to linearize the computation time.

This study proposed approximation-based homomorphic encryption as a replacement for the widely used Brakerski-Gentry-Vaikuntanathan (BFV)-based encryption to reduce the computation resources of blockchain-based encrypted watermarking. The main challenge of our proposed method is the watermarking degradation induced by the approximation approach. Considering the embedding computation comprise several matrices multiplications, the computation error will grow as more computations are applied. This paper mitigated the problem and how to compensate for the error with a higher scaling factor. This study uses Cheon-Kim-Kim-Song (CKKS) encryption [18], [19] as the approximation-based homomorphic encryption. We implemented the proposed system using discrete cosine transform and singular value decomposition (DCT-SVD) watermarking and observed the impact of CKKS encryption on watermarking quality. We prefer the DCT-SVD method for the evaluation due to its simplicity and robustness [20]–[24].

The paper presented the proposed method in section 2 and followed by the evaluation results in section 3. At last, section 4 concludes the paper. The proposed method presents the CKKS homomorphic encryption and its implementation using the DCT-SVD method to provide a secure and provable watermark embedding. The smart contract design winds up the proposed method section. The evaluation section discusses three main topics: i) the resource efficiency of the proposed CKKS-encrypted watermarking compared to existing methods; ii) the quality degradation imposed by the CKKS approximation approach; and iii) the encrypted watermarking robustness under geometrical and filtering attacks.

2. PROPOSED METHOD

This section describes the fundamental literature before presenting the proposed method. The background literature covers CKKS homomorphic encryption and DCT-SVD watermarking. The proposed method presents the CKKS-encrypted watermarking and the smart contract design that implements the system. In summary, Table 1 list all of the notations used in this paper.

Table 1. Mathematical notations used in this study

Symbol	Description
I	Original image
I'	Watermarked image
I''	Tested image (possibly altered watermarked image)
W	Watermark data
W''	Extracted watermark
Sk_A, Pk_A	Authority's secret key and public key
Sk_B, Pk_B	Buyer's secret key and public key
U_I, S_I, Vh_I	Matrices U, S and V^T of an image (I)
U_W, S_W, Vh_W	Matrices U, S and V^T of the watermark image (W)
U_{IX}, S_{IX}, Vh_{IX}	Matrices U, S and V^T of an image (I) encrypted using X's public key
U_{XW}, S_{XW}, Vh_{XW}	Matrices U, S and V^T of the watermark data (W) encrypted using X's public key
S_{XN}	New singular value encrypted using X's public key

2.1. CKKS homomorphic encryption

Homomorphic encryption offers computation over encrypted data without requiring the computing party to decrypt it first. Consequently, it is possible to implement a secure computing platform using a blockchain public ledger by storing the data in an encrypted format. The first generations of fully homomorphic encryption (FHE) [25] use integer encoding. A floating-point number can be encrypted by converting the data into two integers, encoding the decimal part and the fractional one.

Despite its ability to encrypt floating point numbers, it is not suitable for practical floating-point computation. The process demands a heavy load of computing and hefty memory storage. The Cheon-Kim-Kim-Song (CKKS) FHE [18] proposed approximate-based homomorphic encryption that enables fast and lightweight computation over encrypted floating point numbers. Instead of precise integer encoding, the CKKS method uses vectors of complex numbers for faster computation. Nowadays, the CKKS method is commonly used for federated machine learning that outsources data training using a third-party model.

The drawback of CKSS homomorphic encryption is the computation precision. The decrypted value of CKKS encryption will not be exactly equal to the original one due to the approximation method. In vanilla FHE, the result of encrypted addition or multiplication can be decrypted into the same values as explicit computation if the noise budget stands sufficient (1, 2). In contrast, since CKKS FHE implements the learning-with-error (LWE) concept, the encrypted computation will yield an error residue (*e*). The decrypted value is similar to plain computation with *2e* error for addition and *ae* for multiplication (3, 4). The c_0 and c_1 are the ciphertexts of *b*, while C_{mult} is the ciphertext of the multiplication result. Considering the value ***e*** is negligibly small, the approximated result is almost the same as non-encrypted computation.

$$Decryption(Encryption(a) + Encryption(b)) = a + b \quad (1)$$

$$Decryption(Encryption(a) * Encryption(b)) = a * b \quad (2)$$

$$Decryption_{CKKS}\left(Encryption_{CKKS}(a) + Encryption_{CKKS}(b)\right) = a + b + 2e \approx a + b \quad (3)$$

$$Encryption_{CKKS}(a * Encryption_{CKKS}(b)) = (a.c_0, a.c_1) = Cmult$$

$$Decryption_{CKKS}(Cmult) = ab + ae \approx ab \quad (4)$$

Despite the computation error being considerably small, the error from CKKS encryption will propagate to the subsequent computations. In encrypted watermarking cases, the impending processes, inverse transformation, and image reconstruction are complex computations that might upscale the error. This study will investigate the impact of the propagated error on the watermarking quality and the retrievability of the embedded watermark data.

2.2. DCT-SVD watermarking

2.2.1. Watermark embedding

The DCT-SVD watermarking embeds watermark data into the cosines transform domain by using SVD. Here, the DCT-SVD watermarking is preferred due to its proven reliability against various attacks, from geometrical to filtering ones [20], [21]. The detailed step to implement DCT-SVD watermarking is as.

- Transform the original image (*I*) using DCT (5) and decompose the result using SVD operation to retrieve three matrices, the orthogonal matrices U_I and Vh_I and a singular vector S_I (6).

$$c(r,s) = \alpha(r) \cdot \alpha(s) \sum_{x,y=0}^{N-1} \sum \left\{ f(x,y) \cdot cos\left[\frac{(2x+1)\pi r}{2N}\right] \cdot cos\left[\frac{(2y+1)\pi s}{2N}\right] \right\} \quad (5)$$

$$I = U_I S_I V_I^T \quad (6)$$

- Prepare the watermark data (*W*) by decomposing the watermark using SVD to retrieve the singular vector (S_W) (7)

$$W = U_W S_W V_W^T \quad (7)$$

- Apply watermark embedding using one of the following schemes with a scaling factor (*k*). The proposed method runs the embedding processes in an encrypted format, from singular value addition (8) to SVD-based image reconstruction (9).

$$W_D = S_I + k \cdot S_W \quad (8)$$

$$I_{svd} = U_I W_D V_I^T \quad (9)$$

- Run inverse DCT on the SVD-based reconstructed image to form the watermarked image (*I'*) (10).

$$I_{svd} = U_I W_D V_I^T \quad (10)$$

2.2.2. Watermark validation

In this study, we use non-blind watermark extraction to measure the quality degradation caused by encrypted watermarking by comparing it to plain/non-encrypted one. The process is similar to watermark embedding, except that instead of embedding the watermark, we extract the watermark. Later, we compare the singular value with the original image according to the peak signal-to-signal ratio (PSNR), structural similarity index (SSIM), and normalized correlation (NC). The following steps show the extraction process.

- Transform the tested image (I'') using DCT followed by SVD operation to retrieve three decomposed matrices, $U_{I''}$, $S_{I''}$, $Vh_{I''}$ (11).

$$SVD(DCT(I'')) = U_{I''}S_{I''}Vh_{I''} \tag{11}$$

- By referring to the singular vector of the original image (S_I) and the descaling factor (k), extract the watermark's singular vector ($S_{I''}$) (12).

$$S_{W'} = (S_{I''} - S_I)/k \tag{12}$$

- Reconstruct the extracted watermark (W") by using inverse SVD the orthogonal matrices of original image (U_W and Vh_W) with the extracted singular vector ($S_{W'}$) (13).

$$W' = U_W \cdot S_{W'} \cdot V_W^T \tag{13}$$

- Determine the validity of extracted watermark (W') according to the value of PSNR (14), SSIM (15) and NC (16) by comparing it with the original watermark image (W).

$$MSE = \frac{1}{MN}\sum_{X=1}^{N}\sum_{y=1}^{N}\left(W(x,y) - W'(x,y)\right)^2$$

$$PSNR = 10\ \log_{10}\left(\frac{255^2}{MSE}\right) \tag{14}$$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C1)(2\sigma_x\sigma_y + C2)}{(\mu_x^2 + \mu_y^2 + C1)(\sigma_x^2 + \sigma_y^2 + C2)} \tag{15}$$

$$NC = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} W(i,j)\, W'(i,j)}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} W(i,j)}\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} W'(i,j)}} \tag{16}$$

2.3. Blockchain-based watermarking service

The proposed blockchain-based watermarking service uses cognitive-affective-behavior (CAB) model that comprises three interacting actors, the content creators (C), the watermarking authority (A), and the buyer (B) as shown in Figure 1. The content creators generate digital images for sale; thus, they need to imprint unique watermarking into the image for every buyer. The authority (A) uses blockchain to record every licensing transaction between the content creator and the buyers to ensure watermarking accountability and resolve the licensing dispute. The buyer (B) will receive a unique watermarked image with a valid watermark known only to itself. The content creator (C) does not know the watermark data that the authority imbued into the image, while the authority (A) has no clue regarding the original image. The CAB model ensures every actor has access to their respective data and prevent them from issuing valid but illegal watermarked image. The content creator has exclusive access to the original image, the authority knows only the watermarking data, and every buyer has exclusive access to their respective watermarked image.

The use of CKKS homomorphic encryption is to ensure the secrecy of watermarking approval with better efficiency than existing works [16], [17], [25]. The data for watermark embedding is openly available in the blockchain ledger in an encrypted format. It lets anyone securely reproduce the watermark embedding computation. The step-by-step procedure to implement blockchain-based and CKKS-encrypted watermarking services can be broken down into two parts, watermark embedding and watermarking verification as follow.

2.3.1. CKKS-encrypted watermark embedding

Given the original image (I) and watermarking data (W), the encrypted watermarking will generate a unique watermarked image for the buyer (I') securely and transparently using blockchain (BC) and CKKS encryption. As shown in Figure 1 and algorithm 1, the steps for CKKS-encrypted watermarking can be described as follows.

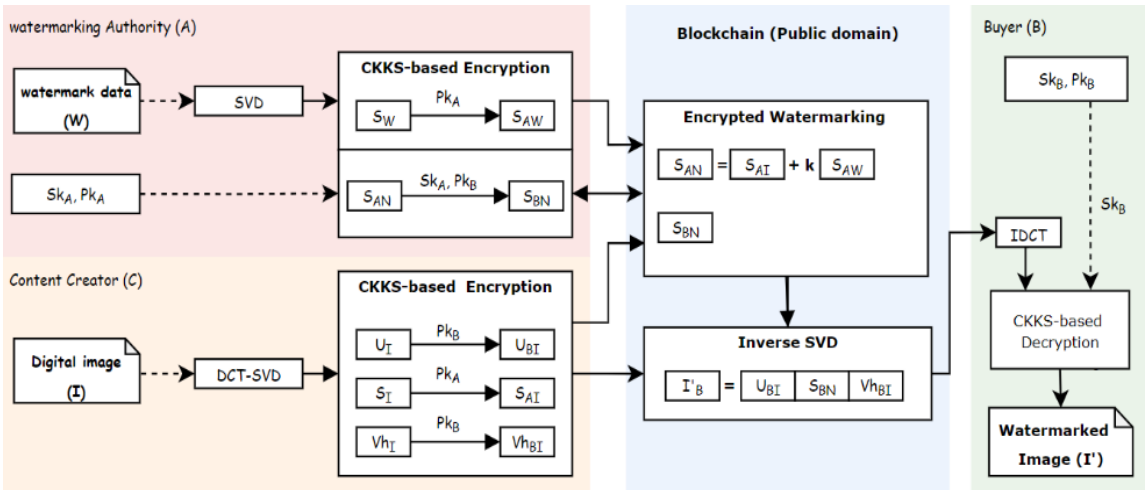

Figure 1. CKKS-encrypted watermark embedding

Algorithm 1. CKKS-encrypted watermark embedding

```
Input: I, W, Pk_A, Pk_B, Sk_A, Sk_B
Output: I'
Actor: C, A, B
C → Advertise_Image(I, PkA, BC)
if ReqB then
      A → Issue_Watermark(W, Pk_A, BC)
      A → Insert_Watermark(Pk_A, Sk_A, Pk_B , BC)
      C → Deliver_Data(Pk_B , BC, CS)
      A → Generate_Result(BC, CS)
      B → Receive_Image(Sk_B , BC, CS)
end if

List of procedures
procedure Advertise_Image(I, Pk_A)
      U_I, S_I, Vh_I ← DCT_SVD(I)
      S_AI ← encrypt(Pk_A, S_I )
      BC ← S_AI
end procedure
procedure Issue_Watermark(W, Pk_A, BC)
      U_W, S_W, Vh_W ← DCT_SVD(W)
      U_AW, S_AW, Vh_AW ← encrypt(Pk_A, U_W , S_W , Vh_W )
      BC ← S_AW
      CS ← U_AW, Vh_AW
      BC ← hash(U_AW, Vh_AW), URL
end procedure
procedure Insert_Watermark(Pk_A, Sk_A, Pk_B, BC)
      S_AN ← S_AI + k × S_AW
      BC ← S_AN
      buffer ← decrypt(S_AN, Sk_A)
      S_BN ← encrypt(Pk_B, buffer)
      BC ← S_BN
end procedure
procedure Deliver_Data(Pk_B, BC, CS)
      U_BI, Vh_BI ← encrypt(Pk_B, U_I, Vh_I)
      CS ← U_BI, Vh_BI
      BC ← hash(U_BI, Vh_BI), URL
end procedure
procedure Generate_Result(BC, CS)
      S_BN ← BC
      U_BI, Vh_BI ← CS
      I'_B ← Inverse_SVD(U_BI, S_BN, Vh_BI)
      CS ← I'_B
      BC ← hash(I'_B), URL
end procedure
procedure Receive Image(SkB , BC, CS)
      I'_B ← verify(BC, CS)
      I'_isvd ← decrypt(Sk_B, I'_B)
      I' ← InverseDCT(I'_isvd)
end procedure
```

- Content creator (C) advertises the image using blockchain (BC). It transforms and decomposes the original image (I) using the DCT-SVD technique into three matrices U_I, S_I, and Vh_I. It will first encrypt the diagonal matrix (S_I) using the authority's public key (Pk_A) into S_{AI} before uploading it to the blockchain.
- Upon receiving a request to buy from the buyer (Req_B), the authority (A) generates unique watermark data and decomposes it using SVD into U_W, S_W, and Vh_W. Next, it will encrypt the data using the public key (Pk_A) into U_{AW}, S_{AW}, Vh_{AW}. The authority uploads the encrypted singular value to the blockchain ledger. Meanwhile, the remaining data U_{AW}, Vh_{AW} are uploaded to the cloud storage.
- The authority then runs watermark embedding by adding the k-scaled watermark data (S_{AW}) into the singular value of the original image (S_{AI}) and generates a new singular value matrix (S_{AN}). This new singular matrix is recorded on the blockchain for transparency. The authority sends the new singular matrix to the buyer via blockchain by first decrypting the S_{AN} and re-encrypted it using the buyer's public key (Pk_B) into S_{BN}.
- Meanwhile, the creator will encrypt the two orthogonal matrices of the original image (U_I, Vh_I) using the buyer's public key (Pk_B) into U_{BI} and Vh_{BI}. The creator will upload the encrypted data into cloud storage (CS) with open access and records the respective URL and hash value to the blockchain ledger.
- The authority generates the embedding result using inverse-SVD (9) to ensure the integrity of the image reconstruction. The computation runs in an encrypted format. The result (I'_B) is stored in cloud storage while its hash value ($hash(I'_B)$) and URL are stored in the blockchain.
- At last, the buyer (B) can verify the correctness of the computation by running step number 5 by itself. If the computation is correct, the buyer can decrypt the result (I'_B) using its secret key (Sk_B). Finally, the buyer runs an inverse DCT operation on the decrypted data to obtain the watermarked image (I').

The proposed method ensures two-sided approval between the buyer and content creator regarding the encrypted watermarking process. If the buyer found out that the authority's watermarking is incorrect, they can reject the transaction. Meanwhile, the content creator might also cancel the transaction if the granted watermark is not unique for every buyer.

2.3.2. CKKS-encrypted watermarking verification

As shown in Figure 2, the proposed method offers two verification tests: originality and validity tests (algorithm 2). The originality test determines whether the tested image (I'') contains the exact information as the original one and has not changed a single bit. In contrast, the validity test concludes whether the tested image contains a valid watermark (W'') granted by the authority, regardless of whether adversaries might have modified the image. The originality test is run by comparing the encrypted data of the tested image ($U_{BI''}$, $S_{AI''}$, and $Vh_{BI''}$) with the stored encrypted data (U_{BI}, S_{AN}, and Vh_{BI}). If the result is the same, it indicates that no one has changed the tested image; thus, its originality is preserved.

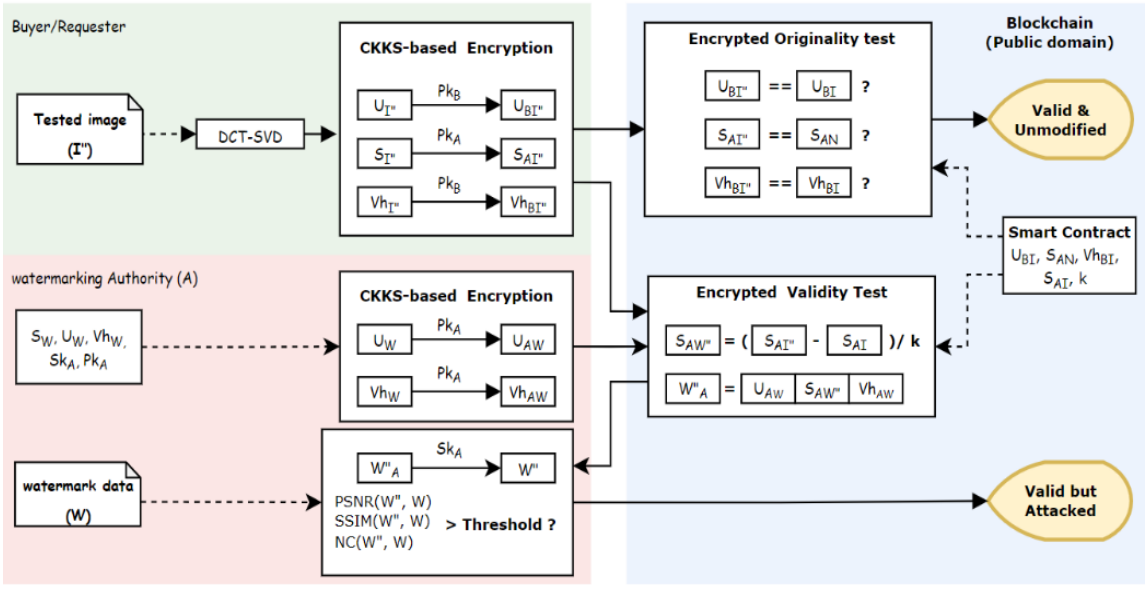

Figure 2. CKKS-encrypted watermark verification

Algorithm 2. CKKS-encrypted watermarking verification

```
Input: I″, Pk_A, Pk_B, Sk_A
Output: Originality of I″, Validity of W″
Actor: A, Any_one
Any_one → Originality_Test(I″, Pk_A, Pk_B , BC, CS)
if Original_status = True then
      I″ is Original and W″ is Valid
else
      I″ has been modified
      A → Extract_Watermark(I″, Pk_A, BC, CS)
      A → Compute_Stats(Sk_A, W″_A)
      if Validity_result = Valid then
          W″ is Valid
      else
          W″ is Invalid
      end if
end if

List of procedures
procedure Originality_Test(I″, Pk_A, Pk_B , BC, CS)
      S_AN ← BC
      U_BI, Vh_BI, U_AW, Vh_AW ← BC, CS
      U_I″, S_I″, Vh_I″ ← DCT_SVD(I″)
      S_AI″ ← encrypt(Pk_A, S_I″)
      U_BI″, Vh_BI″ ← encrypt(Pk_B , U_I″, Vh_I″)
      Original status ← (U_BI = U_BI″) ∧ (S_AN = S_AI″) ∧ (Vh_BI = Vh_BI″)
end procedure
procedure Extract_Watermark(I″, P kA, BC, CS)
      S_AN ← BC
      U_AW , Vh_AW ← BC, CS
      U_I″, S_I″, Vh_I″ ← DCT_SVD(I″)
      S_AI″ ← encrypt(Pk_A, S_I″)
      S_AW″ ← (S_AI″ − S_AN)/k
      W″_A ← Inverse_SVD(U_AW, S_AW″, Vh_AW)
end procedure
procedure Compute_Stats(SkA, W ″A)
      W″ ← decrypt(Sk_A, W″_A)
      ssim ← SSIM(W″, W)
      nc ← NC(W″, W)
      if ssim, nc ≥ threshold then
          Validity result ← Valid
      else
          Validity result ← Invalid
      end if
      BC ← ssim, nc, Validity_result
end procedure
```

The validity test determines the watermark validity by measuring the embedded watermark's quality. If the quality is considerably high (*NC*>0.5), it indicates that the watermark data is present and recoverable by the authority. The authority must run the validity test due to its sole access to the original watermark data. It runs the test by computing PSNR, SSIM, and NC values on the extracted watermark. The requester must provide the extracted watermark data in an encrypted format and store it in the blockchain ledger for public monitoring. The procedure to run the validity test is as follows given the original image (I) and watermarking data (*W*), the encrypted watermarking will generate a unique watermarked.

- At first, the smart contract data is storing the encrypted data from the watermark embedding process, particularly the new singular value of the watermarked image (S_{AN}).
- The requester decomposes the tested image using DCT-SVD and encrypts the singular matrix using the authority's public key (Pk_A) into $S_{AI''}$.
- The requester extracts the embedded watermark using (11, 12, 13) and generate the inverse SVD of it (W''_A).
- The authority computes watermark similarity and its quality by comparing the extracted watermark.
- (W''_A) with the original watermark (W) using SSIM (15) and NC (16) values.
- The authority submits the validation result to the blockchain upon computing the values. If the NC values are greater than the threshold (50%), the watermark is valid; Otherwise, it is invalid.

2.4. Blockchain smart contract

The proposed smart contract consists of five functions: constructor, listing, request, generate, and approval. The constructor function is used to deploy the contract by the authority. The authority must submit

its public key as the function parameter. The listing function acts as the catalog for the content creator to register their digital image.

In buyer activities, the request function sends the buy-out request for the digital image. The buyer must submit their public key as the parameter for the function. Further, the authority uses the generate function to produce the watermarked data and compute a watermark embedding for the respective buyer.

At last, the approval function acts as a state machine that populates the approval from the buyer and content creator regarding the watermarking process. After receiving the approval, the watermarking process is considered valid and the stored encrypted data is referable for watermarking verification. The summary of the functions and their parameters are shown in Table 2.

Table 2. List of smart contract functions

Functions	Sender	Input parameters	Output
Constructor()	Authority	Service name, Authority's public Key	-
Listing()	Content Creator	URL, Hash	Content ID
Request()	Buyer	Content ID, Buyer's public Key, URL, Hash	-
Generate()	Authority	Content ID, URL, Hash	-
Approval()	Content Creator, Buyer	Content ID, approval, URL, Hash	Status

3. RESULTS AND DISCUSSION

For evaluation tests, we use images of a Mandrill (Baboon), Sailboat on Lake (Lake), Male (Man), and Peppers from the SIPI database [26] in grayscale color mode. We use the Tenseal library [19] to implement CKKS homomorphic encryption. As for image quality computations, we use multiple tools from several libraries such as Numpy, Scipy, Scikit-Image, and OpenCV.

3.1. Approximation consequence

Our proposed method must use a considerably high scaling factor due to the approximation technique used by the CKKS homomorphic encryption. This preference is to minimize the propagation error caused by the encrypted watermarking process. In this section, we evaluated the minimum value of the scaling factor that preserves watermarking quality. First, we run the watermark embedding (8) on two different sizes of host images (256 by 256 and 8,192 by 8,192 pixels). After the embedding, we measure the delta between the original image and the watermarked image and compute its standard deviation. The small value of standard deviation indicates that it has insignificant propagation errors. The opposite result means the scaling factor is too small to compensate for the error. Figure 3 shows that the minimum scaling factor must be $\geq 2^{40}$ to yield an unnoticeable propagation error. The results are similar for either a small image (256 by 256 pixels) or a high-resolution one (8,192 by 8,192 pixels). Based on the result, we set 2^{40} as the default scaling factor for our work.

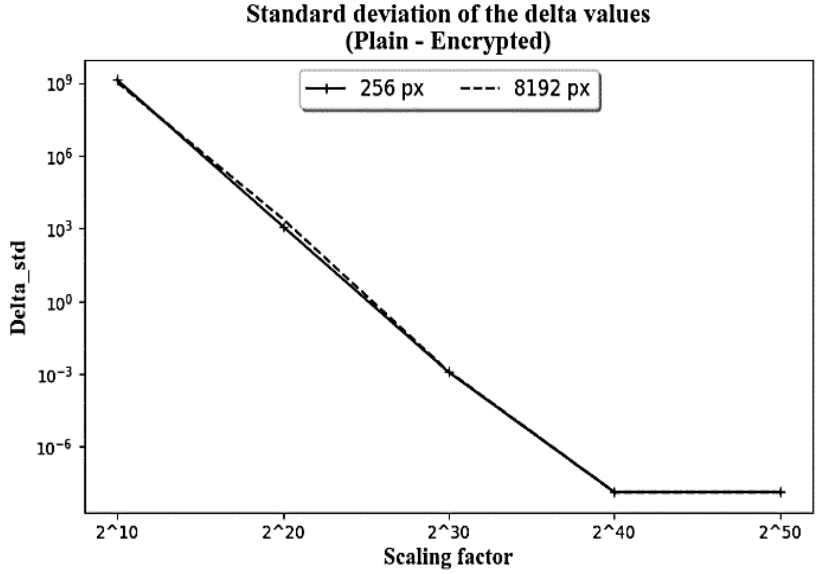

Figure 3. Scaling factor impact on error's standard deviation

3.2. Resource efficiency

Encrypted watermark embedding requires more computation resources than the plain (non-encrypted) embedding process. It uses the extra resources to run: initializing or reading the keys, encrypting the data, running encrypted watermarking, and decrypting the encrypted data. The CKKS-encrypted watermarking has better efficiency than BFV-encrypted watermarking as shown in Figure 4. It is shown by better processor utilization and shorter computation time, up to three times faster. In terms of memory usage can be seen in Figure 4, CKKS-encrypted has a slight advantage over BFV-encrypted watermarking, approximately 200 Megabytes smaller for both sizes of images, 256 by 256 pixels (Baboon, Lake, Pepper) and 512 by 512 pixels (Man).

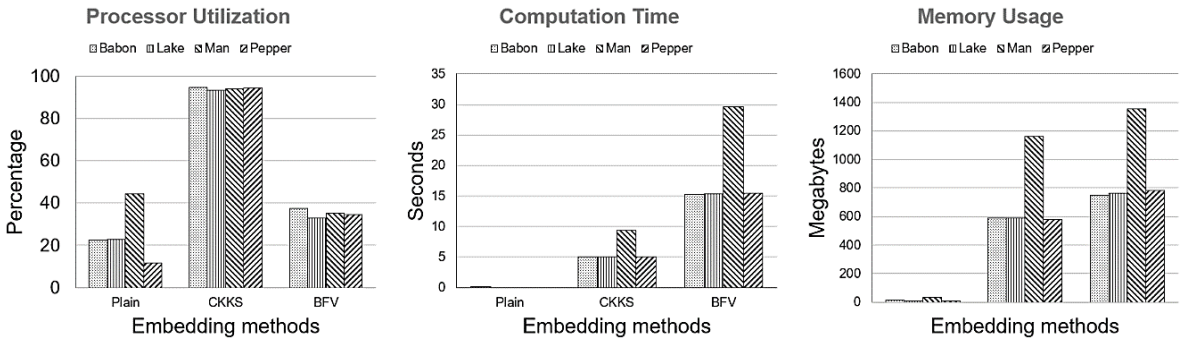

Figure 4. Resource utility comparison between watermark embedding methods

3.3. Quality degradation

According to Figure 5, CKKS-encrypted watermarking has a similar quality to non-encrypted watermark embedding. Meanwhile, the BFV-based encrypted watermarking has significant quality degradation for approximately 25% lower in PSNR value. In addition to lower image quality, BFV-encrypted watermarking has a worse watermark preservation quality. According to Figure 6, BFV-encrypted watermarking suffers in terms of SSIM value. It indicates that the embedding method fails to preserve the structural information of the embedded watermark due to the encryption method. In contrast, the CKKS-encrypted watermarking has a similar SSIM value to the standard, non-encrypted watermarking.

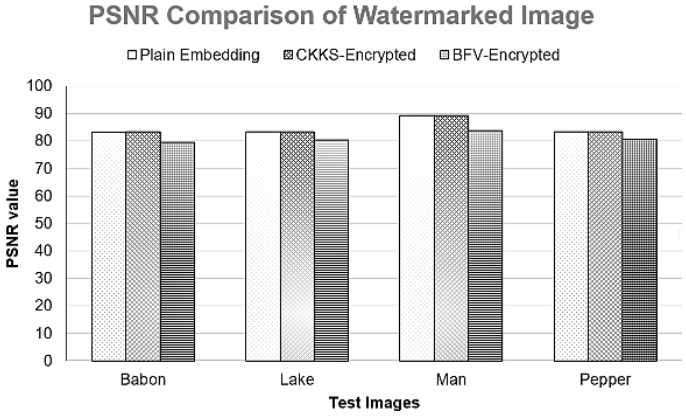

Figure 5. PSNR comparison of watermarked image

The NC value of the proposed method also has a similar pattern to the SSIM one. It has an on-par result with standard embedding. In contrast, the BFV-encrypted watermarking has a worse value of <50%. In short, it is impossible to correlate the embedded watermark with the original one.

3.4. Watermark robustness

This section evaluates the impact of geometrical and filtering attacks on encrypted-watermarking quality. We compared non-encrypted watermarking with CKKS-encrypted watermarking. For the evaluation,

we refer to the three indicators of acclaimed watermarking quality: PSNR, SSIM, and NC. The analysis covers five types of attacks. Two geometrical attacks are rotate 45 degrees, and crop left 25%. The filtering attacks comprise Gaussian blur, Sobel edge filtering, and histogram equalizer.

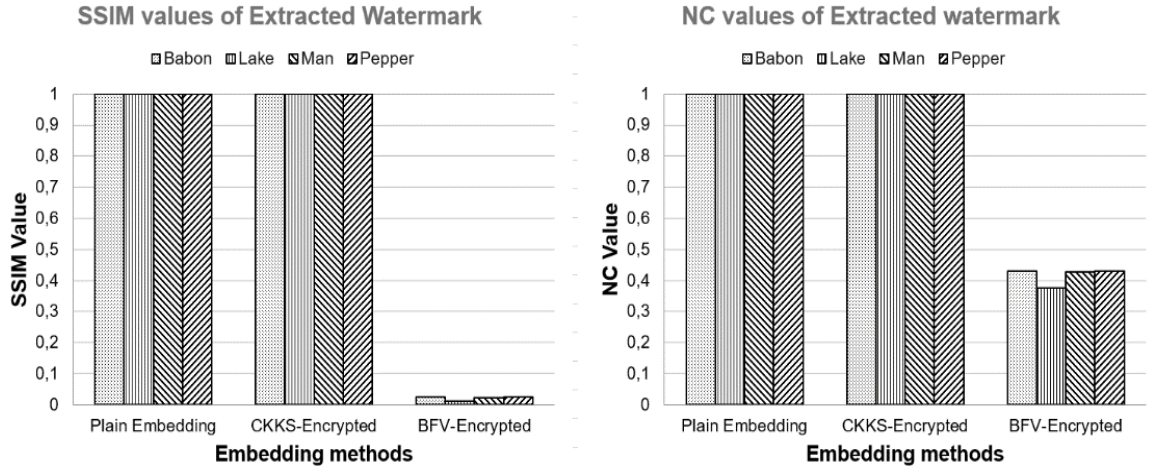

Figure 6. Watermarking quality degradation measured according to SSIM and NC values

Most attacks degrade the PSNR value of the watermarked image significantly. However, the overall watermarking quality (≥46) is higher than the minimum tolerable PSNR value (>30) as shown in Figure 7. In addition, the CKKS-encrypted watermark embedding has similar quality preservation to the non-encrypted one regardless of the attacks. It indicates that the approximation approach has a negligible impact on watermarking quality even under attack conditions. The high-quality preservation results from the 40-bit scaling factor of the CKKS encryption used in the experiment. The setting lets the computation store 40-bit of approximation for the decimal value, which is more than sufficient for SVD-embedding operation.

Referring to Figure 7, CKKS-encrypted watermarking has similar reliability to non-encrypted watermarking, which provides detailed information on PSNR values in Figure 7(a), SSIM comparison in Figure 7(b), and comparison of NC value for the extracted watermarking in Figure 7(c). The value difference is less than 10−6 for both the SSIM and absolute NC values. The results are consistent for all test images. It shows that we can compensate for the approximation technique of CKKS homomorphic encryption by using a high scaling factor.

3.5. Future direction

Our proposed method offers better transparency for blockchain-driven digital rights management (DRM). Most existing works suffer in terms of transparency [11]–[15]. The ones that offer transparency are limited to the data only, by publishing the required data to the cloud or distributed storage [12], [14]. Therefore, we present transparency for data and arithmetic computation of the watermarking process. It lets anyone trace the correctness of the watermark embedding securely without revealing the secret data. As a result, it can preserve the integrity of the watermarking process and the intrinsic value of the watermarked image. The potential application of our proposed method is to ensure the uniqueness of a non-fungible token (NFT) within and beyond the blockchain space.

Our method has a weakness against the histogram equalization attack, as indicated by its low PSNR value. For future works, we suggest exploring other transform functions such as discrete wavelet transform (DWT), hybrid DCT-DWT, or fast Fourier transform (FFT) to improve the robustness of the watermark. A comparative evaluation is necessary to determine which transfer function has the best result in the aspects of robustness against various attacks, resource consumption, and scalability for large images.

For better security, we consider a modular approach by dividing the image into N-by-N parts and computing the watermark embedding respectively for each part. It offers faster encrypted watermarking by outsourcing the watermark embedding to multiple authorities (M). The buyer can apply a hash function to map N^2 parts to M authorities, where N and M are multipliers of 2. This schema offers better secrecy against collusive authorities. Even though all authorities are colluding, it is hard for them to find the correct combination of the M^{N^2} mapping.

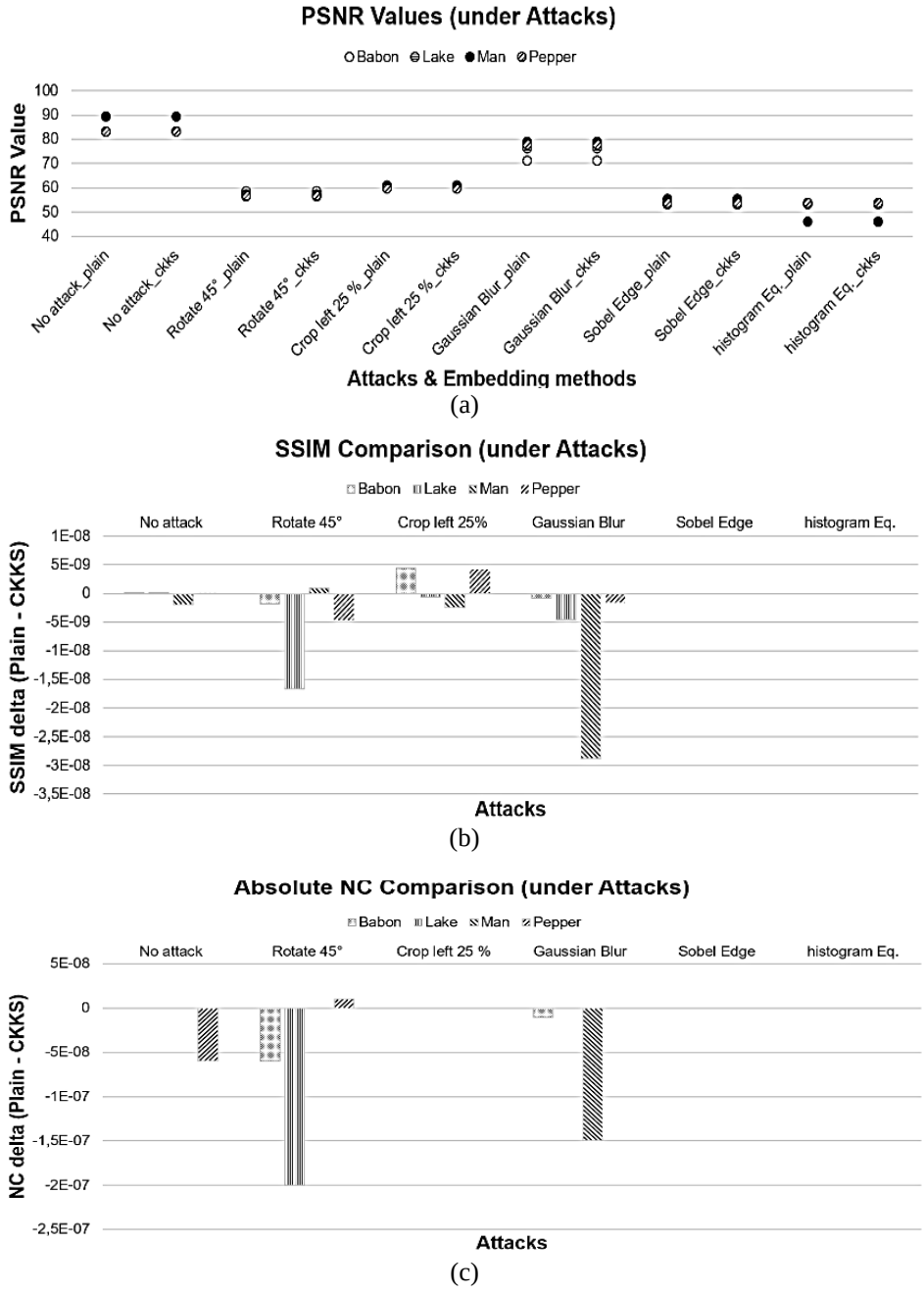

Figure 7. Watermarking quality under various attacks (a) PSNR value of the watermarked image, (b) SSIM value of the extracted watermark, and (c) normalized correlation of the extracted watermark

4. CONCLUSION

This study proved the efficiency of CKKS-encrypted watermarking for blockchain-driven watermarking services. Besides ensuring the secrecy and traceability of the watermarking process, it can preserve watermarking quality similar to plain (non-encrypted) watermarking. In addition, the CKKS-encrypted watermarking has superior quality and efficient resource utilization compared to existing homomorphic encrypted watermarking, particularly the BFV-encrypted one. Under attack conditions, CKKS-encrypted watermarking has similar robustness to the standard (non-encrypted) DCT-SVD watermarking.

REFERENCES

[1] E. Elbasi, "B-DCT based watermarking algorithm for patient data protection in IoMT," in *2020 International Conference on Information Security and Cryptology (ISCTURKEY)*, Dec. 2020, pp. 1–4, doi: 10.1109/ISCTURKEY51113.2020.9307963.

[2] A. Anand, A. K. Singh, and H. Zhou, "ViMDH: visible-imperceptible medical data hiding for internet of medical things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 849–856, Jan. 2023, doi: 10.1109/TII.2022.3172622.
[3] P. Aparna and P. V. V. Kishore, "A blind medical image watermarking for secure e-healthcare application using crypto-watermarking system," *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 1558–1575, Aug. 2019, doi: 10.1515/jisys-2018-0370.
[4] J.-C. Huang, F.-G. Jeng, and T.-H. Chen, "A new buyer-seller watermarking protocol without multiple watermarks insertion," *Multimedia Tools and Applications*, vol. 76, no. 7, pp. 9667–9679, Apr. 2017, doi: 10.1007/s11042-016-3573-1.
[5] F. Frattolillo, "A multiparty watermarking protocol for cloud environments," *Journal of Information Security and Applications*, vol. 47, pp. 246–257, Aug. 2019, doi: 10.1016/j.jisa.2019.05.011.
[6] C. Song, J. Sang, and S. Sudirman, "A buyer-seller watermarking protocol for digital secondary market," *Multimedia Tools and Applications*, vol. 77, no. 1, pp. 225–249, Jan. 2018, doi: 10.1007/s11042-016-4247-8.
[7] S. Katzenbeisser, A. Lemma, M. U. Celik, M. van der Veen, and M. Maas, "A buyer–seller watermarking protocol based on secure embedding," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 783–786, Dec. 2008, doi: 10.1109/TIFS.2008.2002939.
[8] C.-C. Chang, H.-C. Tsai, and Y.-P. Hsieh, "An efficient and fair buyer–seller fingerprinting scheme for large scale networks," *Computers and Security*, vol. 29, no. 2, pp. 269–277, Mar. 2010, doi: 10.1016/j.cose.2009.08.008.
[9] Z. Xu, A. Li, and H. Gao, "Bandwidth efficient buyer-seller watermarking protocol," *International Journal of Information and Computer Security*, vol. 5, no. 1, 2012, doi: 10.1504/IJICS.2012.051079.
[10] T. Bianchi and A. Piva, "TTP-free asymmetric fingerprinting based on client side embedding," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1557–1568, Oct. 2014, doi: 10.1109/TIFS.2014.2340581.
[11] F. Frattolillo, "A watermarking protocol based on blockchain," *Applied Sciences*, vol. 10, no. 21, Nov. 2020, doi: 10.3390/app10217746.
[12] C. Song, H. Wang, W. Zhang, S. Sudirman, and H. Zhu, "A blockchain based buyer-seller watermark protocol with trustless third party," *Recent Advances in Electrical and Electronic Engineering (Formerly Recent Patents on Electrical and Electronic Engineering)*, vol. 13, no. 6, pp. 942–950, Nov. 2020, doi: 10.2174/2352096513999200623121213.
[13] A. Qureshi and D. M. Jiménez, "Blockchain-based multimedia content protection: review and open challenges," *Applied Sciences*, vol. 11, no. 1, Dec. 2020, doi: 10.3390/app11010001.
[14] F. Frattolillo, "Blockchain and cloud to overcome the problems of buyer and seller watermarking protocols," *Applied Sciences*, vol. 11, no. 24, Dec. 2021, doi: 10.3390/app112412028.
[15] B. Wu, Y. Peng, and C. Wang, "A TTP watermarking protocol based on visual cryptography," *Multimedia Tools and Applications*, vol. 81, no. 28, pp. 41079–41101, Nov. 2022, doi: 10.1007/s11042-022-13002-y.
[16] A. Basuki, I. Setiawan, and D. Rosiyadi, "Preserving privacy for blockchain-driven image watermarking using fully homomorphic encryption," in *The 2021 International Conference on Computer, Control, Informatics and Its Applications*, Oct. 2021, pp. 151–155, doi: 10.1145/3489088.3489130.
[17] A. I. Basuki, I. Setiawan, and D. Rosiyadi, "Improving efficiency on BFV--based encrypted watermarking using Hadamard product decomposition," NISS, 2022.
[18] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10624, Springer International Publishing, 2017, pp. 409–437.
[19] A. Benaissa, B. Retiat, B. Cebere, and A. E. Belfedhal, "TenSEAL: a library for encrypted tensor operations using homomorphic encryption," *arXiv preprint arXiv:2104.03152*, 2021.
[20] A. Sverdlov, S. Dexter, and A. M. Eskicioglu, "Robust DCT-SVD domain image watermarking for copyright protection: Embedding data in all frequencies," in *2005 13th European Signal Processing Conference*, 2005, pp. 1–4.
[21] S.-J. Horng, D. Rosiyadi, P. Fan, X. Wang, and M. K. Khan, "An adaptive watermarking scheme for e-government document images," *Multimedia Tools and Applications*, vol. 72, no. 3, pp. 3085–3103, Oct. 2014, doi: 10.1007/s11042-013-1579-5.
[22] F. Ernawan and D. Ariatmanto, "Image watermarking based on integer wavelet transform-singular value decomposition with variance pixels," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 3, pp. 2185–2195, Jun. 2019, doi: 10.11591/ijece.v9i3.pp2185-2195.
[23] S. A. H. Nair and P. Aruna, "Comparison of DCT, SVD and BFOA based multimodal biometric watermarking systems," *Alexandria Engineering Journal*, vol. 54, no. 4, pp. 1161–1174, Dec. 2015, doi: 10.1016/j.aej.2015.07.002.
[24] B. A. Sultan and L. E. George, "Color image compression based on spatial and magnitude signal decomposition," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 5, pp. 4069–4081, Oct. 2021, doi: 10.11591/ijece.v11i5.pp4069-4081.
[25] H. Chen, K. Laine, and R. Player, "Simple encrypted arithmetic library - SEAL v2.1," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10323, 2017, pp. 3–18.
[26] USC, "Volume 3: miscellaneous," Signal and Image Processing Institute, Ming Hsieh Department of Electrical and Computer Enggineering. https://sipi.usc.edu/database/database.php?volume=misc (accessed Sep. 15, 2021).

BIOGRAPHIES OF AUTHORS

Didi Rosiyadi is a senior researcher at the Research center for Data and Information Sciences, National Research and Innovation Agency (BRIN). He is currently Acting Director of Measurement and Indicators for Research, Technology, and Innovation, BRIN. He received his Ph.D. degree from the National Taiwan University of Science and Technology in Information Security. His research interests include information systems, information security, multimedia watermarking, and blockchain. He has been a visiting researcher in various research centers and universities worldwide, from CERN-Switzerland and The University of Zagreb - Croatia, to several universities in Italia, Taiwan, Thailand, and Bangladesh. He can be contacted at email: didi.rosiyadi@brin.go.id.

Akbari Indra Basuki is a researcher at Research Center for Data and Information Sciences, National Research and Innovation Agency (BRIN). He holds a magister of sciences from Bandung Institute of Technology, majoring in Computer Engineering. His research interests include data security, blockchain, network security, and programmable networks. He is actively involved in research activities by acting as a TPC member for several international conferences such as ACM IC3INA and NISS 2022. He was a visiting researcher at KMUTT–Thailand. In addition, he is currently a visiting lecturer at the University of Padjadjaran. He can be contacted at email: akba002@brin.go.id.

Taufik Iqbal Ramdhani is a researcher at Data and Information Science Research Center. He holds a bachelor's and master's degree in cybersecurity from Maranatha Christian University and Florida International University, USA. His research interests include blockchain and public key infrastructure. He actively joined the research and development team of the digital signature implementation in iOTENTIK-BPPT since 2015. He also won several hackathon events in his home institution. As an IT auditor, he joined Indonesia's national selection committee for public service recruitment. He was also involved in securing the 2019 general election system. He can be reached at tauf022@brin.go.id.

Heru Susanto is currently as head of Data Security Research Group, Research Center for Data and Information Sciences, National Research Innovation Agency, Indonesia. At present he is an Honorary Professor, Department of Information Management, College of Management, Tunghai University, Taichung, Taiwan, and also Visiting Professor at Research Center for Innovative Engineering, University of Technology Brunei. Dr. Susanto has worked as an IT professional in several roles, including Web Division Head of IT Strategic, and Prince Muqrin Chair for Information Security Technologies. His research interests are in the areas of information security, 5G technologies, grid application, big data, business process re-engineering, and e-marketing. Dr. Susanto received a B.Sc. in Computer Science; an MBA in Marketing Management; an MSc in Computer Sciences; and a Ph.D. in Information Security from IPB University, IPMI Business School, King Saud University, and Tunghai University respectively. Nowadays, Dr. Heru Successfully authoring 8 Books published by Francis and Taylor Group, such as: Information Security Management Systems, Business Process Reengineering an ICT Approach, The Emerging Technology of Big Data, Human Capital through ICT, Chemical Technology and Informatics in Chemistry with Applications. In other hand, he also has more than 135 peer-reviewed publication within journals, proceeding and book chapter. He can be contacted at heru.susanto@utb.edu.bn or heru015@brin.go.id.

Yusnan Hasani Siregar is a researcher at Research Center for Appropriate Technology, National Research and Innovation Agency (BRIN). He received a magister of sciences from Bandung Institute of Technology, Indonesia. His research interests are related to IoT, automation, and image processing for agriculture productivity. He can be contacted at yusn003@brin.go.id.